Numerical Stability of (Randomized) methods

Yuji Nakatsukasa
Oxford University

MORAT Lecture 3/3

Numerical stability: see book [Higham 02]

Question: Can a computed result be trusted?

e.g. is $Ax = b$ always solved correctly via the LU algorithm?

Numerical stability: see book [Higham 02]

Question: Can a computed result be trusted?

e.g. is $Ax = b$ always solved correctly via the LU algorithm?

- The situation is complicated. For example, let $A = U\Sigma V^T$, where $U = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $\Sigma = \begin{bmatrix} 1 & \\ & 10^{-15} \end{bmatrix}$, $V = I$, and let $b = A\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ (i.e., $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$).

Numerical stability: see book [Higham 02]

Question: Can a computed result be trusted?

e.g. is $Ax = b$ always solved correctly via the LU algorithm?

- The situation is complicated. For example, let $A = U\Sigma V^T$, where $U = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $\Sigma = \begin{bmatrix} 1 & \\ & 10^{-15} \end{bmatrix}$, $V = I$, and let $b = A\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ (i.e., $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$).

  In MATLAB, `x = A\b` outputs $\begin{bmatrix} 1.0000 \\ 0.94206 \end{bmatrix}$

- Did something go wrong?

Numerical stability: see book [Higham 02]

Question: Can a computed result be trusted?

e.g. is $Ax = b$ always solved correctly via the LU algorithm?

- The situation is complicated. For example, let $A = U\Sigma V^T$, where $U = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $\Sigma = \begin{bmatrix} 1 & \\ & 10^{-15} \end{bmatrix}$, $V = I$, and let $b = A\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ (i.e., $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$).

  In MATLAB, `x = A\b` outputs $\begin{bmatrix} 1.0000 \\ 0.94206 \end{bmatrix}$

- Did something go wrong?

  NO—this is a ramification of ill-conditioning, not instability

- In fact, $\|\mathtt{Ax} - \mathtt{b}\|_2 (= \|A\hat{x} - b\|_2) \approx 10^{-16}$

(After this hour, make sure you can explain what happened above!)

Conditioning and stability

- Conditioning: sensitivity of a problem (e.g. of finding $y = f(x)$ given x) to perturbation in inputs.

$$\text{Condition number} \quad \kappa := \lim_{\|\delta x\|_2 \to 0} \sup_{\delta x} \frac{\|f(x+\delta x) - f(x)\|}{\|f(x)\|} \Big/ \frac{\|\delta x\|}{\|x\|}$$

(sometimes absolute condition number:
$\lim_{\Delta x \to 0} \sup_{\Delta x} \|f(x+\Delta x) - f(x)\| / \|\Delta x\|$)

- Backward Stability: property of algorithm
Describes if the computed solution $\hat{y}$ is exact solution of a nearby input, that is, $\hat{y} = f(x + \Delta x)$ for a small Δx.

Conditioning and stability

- Conditioning: sensitivity of a problem (e.g. of finding $y = f(x)$ given x) to perturbation in inputs.

$$\text{Condition number} \quad \kappa := \lim_{\|\delta x\|_2 \to 0} \sup_{\delta x} \frac{\|f(x+\delta x) - f(x)\|}{\|f(x)\|} \Big/ \frac{\|\delta x\|}{\|x\|}$$

  (sometimes absolute condition number:
  $\lim_{\Delta x \to 0} \sup_{\Delta x} \|f(x+\Delta x) - f(x)\| / \|\Delta x\|$)

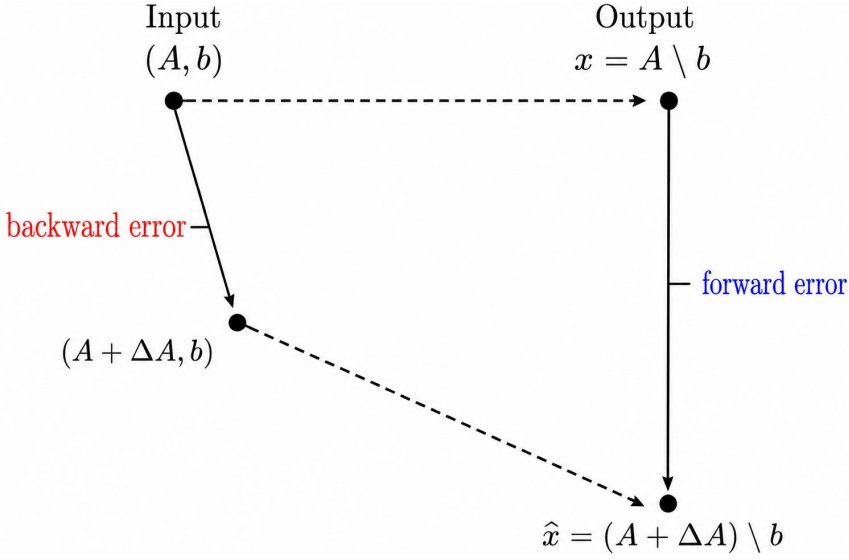
- Backward Stability: property of algorithm
  Describes if the computed solution $\hat{y}$ is exact solution of a nearby input, that is, $\hat{y} = f(x + \Delta x)$ for a small Δx.

If problem is ill-conditioned $\kappa \gg 1$, then blame the **problem**, not the **algorithm**

Notation/convention: $\hat{x}$ is a computed approximation to x (e.g. of $x = A^{-1}b$)
ϵ is $O(u)$; in stability analysis we write e.g. $u, 10u, (m+n)u, mnu$ all as ϵ

- Hence norm choice does not matter here

Forward error and backward error

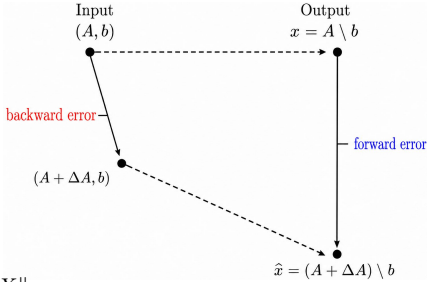


Numerical stability: backward stability

For computational task $\underbrace{Y}_{\text{output}} = f(\underbrace{X}_{\text{input}})$,

let $\hat{Y}$ be computed approximant

- Ideally, error $\|Y - \hat{Y}\|/\|Y\| = \epsilon$: seldom true
  (u: unit roundoff, $\approx 10^{-16}$ in standard double precision)
- Good alg is Backward stable $\hat{Y} = f(X + \Delta X)$, $\frac{\|\Delta X\|}{\|X\|} = \epsilon$
  "exact solution of slightly wrong input "

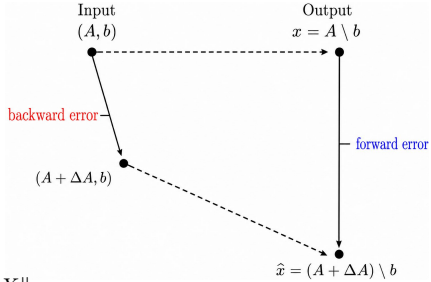


Numerical stability: backward stability

For computational task $\underbrace{Y}_{\text{output}} = f(\underbrace{X}_{\text{input}})$,

let $\hat{Y}$ be computed approximant

- Ideally, error $\|Y - \hat{Y}\|/\|Y\| = \epsilon$: seldom true
  (u: unit roundoff, $\approx 10^{-16}$ in standard double precision)
- Good alg is Backward stable $\hat{Y} = f(X + \Delta X)$, $\frac{\|\Delta X\|}{\|X\|} = \epsilon$
  "exact solution of slightly wrong input "



- Justification: **Input (matrix) is usually inexact anyway!** $f(X + \Delta X)$ is just as good at $f(X)$ at approximating $f(X_*)$, where $\|\Delta X\| = O(\epsilon\|X - X_*\|)$
  But it may not mean $\hat{Y} - Y$ is small if ill-conditioned $\kappa \gg 1$

- Forward stability $\|Y - \hat{Y}\|/\|Y\| = O(\kappa\epsilon)$ "error is as small as backward stable alg." (sometimes used to mean small error; we follow Higham's book [2002])

Backward stable+well conditioned=accurate solution

Suppose

- $Y = f(X)$ computed backward stably i.e., $\hat{Y} = f(X + \Delta X)$, $\|\Delta X\| = \epsilon$.

Then with $\kappa = \lim_{\|\Delta x\|_2 \to 0} \sup_{\Delta x} \frac{\|f(X) - f(X+\Delta X)\|}{\|\Delta X\|}$,

$$\|\hat{Y} - Y\| \lesssim \kappa\epsilon$$

(relative version possible)

Backward stable+well conditioned=accurate solution

Suppose

- $Y = f(X)$ computed backward stably i.e., $\hat{Y} = f(X + \Delta X)$, $\|\Delta X\| = \epsilon$.

Then with $\kappa = \lim_{\|\Delta x\|_2 \to 0} \sup_{\Delta x} \frac{\|f(X) - f(X+\Delta X)\|}{\|\Delta X\|}$,

$$\|\hat{Y} - Y\| \lesssim \kappa\epsilon$$

(relative version possible) 'proof':

$$\|\hat{Y} - Y\| = \|f(X + \Delta X) - f(X)\| \lesssim \kappa\|\Delta X\|\|f(X)\| = \kappa\epsilon$$

Backward stable+well conditioned=accurate solution

Suppose

- $Y = f(X)$ computed backward stably i.e., $\hat{Y} = f(X + \Delta X)$, $\|\Delta X\| = \epsilon$.

Then with $\kappa = \lim_{\|\Delta x\|_2 \to 0} \sup_{\Delta x} \frac{\|f(X) - f(X+\Delta X)\|}{\|\Delta X\|}$,

$$\|\hat{Y} - Y\| \lesssim \kappa\epsilon$$

(relative version possible) 'proof':

$$\|\hat{Y} - Y\| = \|f(X + \Delta X) - f(X)\| \lesssim \kappa\|\Delta X\|\|f(X)\| = \kappa\epsilon$$

If well-conditioned $\kappa = O(1)$, good accuracy! Important examples:

- Well-conditioned linear system $Ax = b$, $\kappa_2(A) \approx 1$
- Eigenvalues of symmetric matrices (via Weyl's bound $\lambda_i(A + E) \in \lambda_i(A) + [-\|E\|_2, \|E\|_2]$)
- Singular values of any matrix $\sigma_i(A + E) \in \sigma_i(A) + [-\|E\|_2, \|E\|_2]$

Note: eigvecs/singvecs can be highly ill-conditioned

Matrix condition number

$$\kappa_2(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} (\geq 1)$$

e.g. for linear systems. (when A is $m \times n (m > n)$, $\kappa_2(A) = \frac{\sigma_1(A)}{\sigma_n(A)}$) A backward stable soln for $Ax = b$, s.t. $(A + \Delta A)\hat{x} = b$ satisfies, assuming $\|\Delta A\| \leq \epsilon \|A\|$ and $\kappa_2(A) \ll \epsilon^{-1}$, (so $\|A^{-1}\Delta A\| \ll 1$),

$$\frac{\|\hat{x} - x\|}{\|x\|} \lesssim \epsilon \kappa_2(A)$$

Matrix condition number

$$\kappa_2(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} (\geq 1)$$

e.g. for linear systems. (when A is $m \times n (m > n)$, $\kappa_2(A) = \frac{\sigma_1(A)}{\sigma_n(A)}$) A backward stable soln for $Ax = b$, s.t. $(A + \Delta A)\hat{x} = b$ satisfies, assuming $\|\Delta A\| \leq \epsilon \|A\|$ and $\kappa_2(A) \ll \epsilon^{-1}$, (so $\|A^{-1}\Delta A\| \ll 1$),

$$\frac{\|\hat{x} - x\|}{\|x\|} \lesssim \epsilon \kappa_2(A)$$

'proof': By Neumann series

$$(A + \Delta A)^{-1} = (A(I + A^{-1}\Delta A))^{-1} = (I - A^{-1}\Delta A + O(\|A^{-1}\Delta A\|^2))A^{-1}$$

So $\hat{x} = (A + \Delta A)^{-1}b = A^{-1}b - A^{-1}\Delta A A^{-1}b + O(\|A^{-1}\Delta A\|^2) =$
$x - A^{-1}\Delta A x + O(\|A^{-1}\Delta A\|^2)$, Hence

$$\|x - \hat{x}\| \lesssim \|A^{-1}\Delta A x\| \leq \|A^{-1}\|\|\Delta A\|\|x\| \leq \epsilon \|A\|\|A^{-1}\|\|x\| = \epsilon \kappa_2(A)\|x\|$$

Backward stability of triangular systems

Recall $Ax = b$ via pivoted LU $A = PLU$: $Ly = P^T b$, $Ux = y$ (triangular systems).
The computed solution $\hat{x}$ for a (upper/lower) triangular linear system $Rx = b$ solved via back/forward substitution is backward stable, i.e.,

$$(R + \Delta R)\hat{x} = b, \qquad \|\Delta R\| = O(\epsilon \|R\|).$$

Proof: [Trefethen-Bau 98] or [Higham 02]

- backward error can be bounded componentwise
- this means $\|\hat{x} - x\|/\|x\| \leq \epsilon \kappa_2(R)$
  - (unavoidably) poor worst-case (and attainable) bound when ill-conditioned
  - often better with triangular systems

(In)stability of $Ax = b$ via LU with pivots

Fact Computed $\hat{L}\hat{U}$ satisfies $\frac{\|\hat{L}\hat{U}-A\|}{\|L\|\|U\|} = \epsilon$

(note: not $\frac{\|\hat{L}\hat{U}-A\|}{\|A\|} = \epsilon$)

- If $\|L\|\|U\| = O(\|A\|)$, then $(L + \Delta L)(U + \Delta U)\hat{x} = b$
  $\Rightarrow \hat{x}$ backward stable solution (exercise)

(In)stability of $Ax = b$ via LU with pivots

Fact Computed $\hat{L}\hat{U}$ satisfies $\frac{\|\hat{L}\hat{U}-A\|}{\|L\|\|U\|} = \epsilon$

(note: not $\frac{\|\hat{L}\hat{U}-A\|}{\|A\|} = \epsilon$)

- If $\|L\|\|U\| = O(\|A\|)$, then $(L + \Delta L)(U + \Delta U)\hat{x} = b$
  $\Rightarrow \hat{x}$ backward stable solution (exercise)

Question: Does $LU = A + \Delta A$ or $LU = PA + \Delta A$ with $\|\Delta A\| = \epsilon\|A\|$ hold?

Without pivot ($P = I$): $\|L\|\|U\| \gg \|A\|$ unboundedly (e.g. $\left[\begin{smallmatrix} \epsilon & 1 \\ 1 & 1 \end{smallmatrix}\right]$) unstable

(In)stability of $Ax = b$ via LU with pivots

Fact Computed $\hat{L}\hat{U}$ satisfies $\frac{\|\hat{L}\hat{U}-A\|}{\|L\|\|U\|} = \epsilon$

(note: not $\frac{\|\hat{L}\hat{U}-A\|}{\|A\|} = \epsilon$)

- If $\|L\|\|U\| = O(\|A\|)$, then $(L + \Delta L)(U + \Delta U)\hat{x} = b$
  $\Rightarrow \hat{x}$ backward stable solution (exercise)

Question: Does $LU = A + \Delta A$ or $LU = PA + \Delta A$ with $\|\Delta A\| = \epsilon\|A\|$ hold?

Without pivot ($P = I$): $\|L\|\|U\| \gg \|A\|$ unboundedly (e.g. $\left[\begin{smallmatrix} \epsilon & 1 \\ 1 & 1 \end{smallmatrix}\right]$) unstable

With pivots:

- Worst-case: $\|L\|\|U\| \gg \|A\|$ grows exponentially with n, unstable
  - growth governed by that of $\|L\|\|U\|/\|A\| \Rightarrow \|U\|/\|A\|$
- In practice (average case): perfectly stable
  - Hence this is how $Ax = b$ is solved, despite alternatives with guaranteed stability exist (but slower; e.g. via SVD, or QR (next))

Resolution/explanation: among biggest open problems in numerical linear algebra!

Stability of Cholesky for $A \succ 0$

Cholesky $A = R^T R$ for $A \succ 0$

- succeeds without pivot (active matrix is always positive definite)
- R never contains entries $> \sqrt{\|A\|_2}$

$$A = \underbrace{\begin{bmatrix} * \\ * \\ * \\ * \\ * \end{bmatrix} \begin{bmatrix} * & * & * & * & * \end{bmatrix}}_{R_1 R_1^T} + \underbrace{\begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix}}_{\text{also PSD}}$$

(exercise: show $\|R_1\|_2 \leq \sqrt{\|A\|_2}$)

$\Rightarrow$ backward stable! Hence positive definite linear system $Ax = b$ stable via Cholesky

(In)stability of QR factorization

Gram-Schmidt has subtle stability

- plain (classical) version: $\|\hat{Q}^T\hat{Q} - I\| \leq \epsilon(\kappa_2(A))^2$
- modified Gram-Schmidt (orthogonalize 'one vector at a time'): $\|\hat{Q}^T\hat{Q} - I\| \leq \epsilon\kappa_2(A)$
- Classical Gram-Schmidt twice (G-S again on computed $\hat{Q}$): $\|\hat{Q}^T\hat{Q} - I\| \leq \epsilon$

CholeskyQR: $A^TA = R^TR$, $Q = AR^{-1}$ is unstable as is, $\kappa_2(\hat{Q}) \approx u(\kappa_2(A))^2$

- Iterative refinement V powerful: do CholeskyQR on $\hat{Q}$
  [Yamamoto-N.-Yanagisawa-Fukaya 15]
- Randomized CholeskyQR: First sketch $SA = Q_1R_1$, $A_1 = AR_1^{-1}$, then CholeskyQR on A_1 is stable [Balabanov 22]

Stability of Householder QR

With Householder QR, the computed $\hat{Q}, \hat{R}$ satisfy

$$\|\hat{Q}^T\hat{Q} - I\| = O(\epsilon), \quad \|A - \hat{Q}\hat{R}\| = O(\epsilon\|A\|),$$

and (of course) R upper triangular.

Rough proof

- Each reflector orthogonal, so satisfies $fl(H_iA) = H_iA + \epsilon_i\|A\|$
- Hence $(\hat{R} =) fl(H_n \cdots H_1 A) = H_n \cdots H_1 A + \epsilon\|A\|$
- $fl(H_n \cdots H_1) =: \hat{Q}^T = H_n \cdots H_1 + \epsilon$, thus $\hat{Q}\hat{R} = A + \epsilon\|A\|$

Stability of Householder QR

With Householder QR, the computed $\hat{Q}, \hat{R}$ satisfy

$$\|\hat{Q}^T\hat{Q} - I\| = O(\epsilon), \quad \|A - \hat{Q}\hat{R}\| = O(\epsilon\|A\|),$$

and (of course) R upper triangular.

Rough proof

- Each reflector orthogonal, so satisfies $fl(H_iA) = H_iA + \epsilon_i\|A\|$
- Hence $(\hat{R} =) fl(H_n \cdots H_1 A) = H_n \cdots H_1 A + \epsilon\|A\|$
- $fl(H_n \cdots H_1) =: \hat{Q}^T = H_n \cdots H_1 + \epsilon$, thus $\hat{Q}\hat{R} = A + \epsilon\|A\|$

Notes:

- This doesn't mean $\|\hat{Q} - Q\|, \|\hat{R} - R\|$ are small at all! Indeed Q, R are as ill-conditioned as A, but **product QR is not**

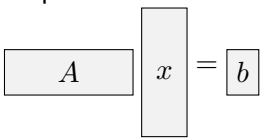
QR-based soln for $Ax = b$, overdetermined (least squares), underdetermined systems

- $Ax = b$ solved by $A = QR$, $x = R^{-1}Q^T b$ is backward stable (arument LU but now obviously $\|\hat{Q}\|\|\hat{R}\| \approx \|A\|$)

QR-based soln for $Ax = b$, overdetermined (least squares), underdetermined systems

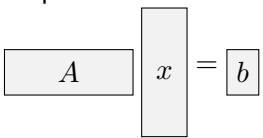
- $Ax = b$ solved by $A = QR$, $x = R^{-1}Q^Tb$ is backward stable (arument LU but now obviously $\|\hat{Q}\|\|\hat{R}\| \approx \|A\|$)
- Overdetermined $\min_x \|Ax - b\|_2$: solved by $A = QR$, $x = R^{-1}Q^Tb$ is backward stable (NB normal eqn $A^TAx =$ is NOT)

QR-based soln for $Ax = b$, overdetermined (least squares), underdetermined systems

- $Ax = b$ solved by $A = QR$, $x = R^{-1}Q^Tb$ is backward stable (arument LU but now obviously $\|\hat{Q}\|\|\hat{R}\| \approx \|A\|$)
- Overdetermined $\min_x \|Ax - b\|_2$: solved by $A = QR$, $x = R^{-1}Q^Tb$ is backward stable (NB normal eqn $A^TAx =$ is NOT)
- Underdetermined 

  which solution?

QR-based soln for $Ax = b$, overdetermined (least squares), underdetermined systems

- $Ax = b$ solved by $A = QR$, $x = R^{-1}Q^T b$ is backward stable (arument LU but now obviously $\|\hat{Q}\|\|\hat{R}\| \approx \|A\|$)
- Overdetermined $\min_x \|Ax - b\|_2$: solved by $A = QR$, $x = R^{-1}Q^T b$ is backward stable (NB normal eqn $A^T Ax =$ is NOT)
- Underdetermined 

  which solution?

  often minimum-norm soln sought: $\min \|x\|_2$ subject to $Ax = b$

  QR-based soln $A^T = QR$, $x = Q(R^{-1}b)$ gives it, backward stable

Orthogonal Linear Algebra

With orthogonal matrices Q,

$$\frac{\|fl(QA) - QA\|}{\|QA\|} \leq \epsilon, \qquad \frac{\|fl(AQ) - AQ\|}{\|AQ\|} \leq \epsilon$$

Orthogonal Linear Algebra

With orthogonal matrices Q,

$$\frac{\|fl(QA) - QA\|}{\|QA\|} \leq \epsilon, \qquad \frac{\|fl(AQ) - AQ\|}{\|AQ\|} \leq \epsilon$$

whereas in general, $\|fl(AB) - AB\| \leq \epsilon \|A\| \|B\|$, so
$\|fl(AB) - AB\| / \|AB\| \leq \epsilon \min(\kappa_2(A), \kappa_2(B))$

Orthogonal Linear Algebra

With orthogonal matrices Q,

$$\frac{\|fl(QA) - QA\|}{\|QA\|} \leq \epsilon, \qquad \frac{\|fl(AQ) - AQ\|}{\|AQ\|} \leq \epsilon$$

whereas in general, $\|fl(AB) - AB\| \leq \epsilon \|A\| \|B\|$, so
$\|fl(AB) - AB\| / \|AB\| \leq \epsilon \min(\kappa_2(A), \kappa_2(B))$

Hence algorithms using orthogonal matrices are usually stable whereas those involving ill-conditioned matrices are often unstable (but not always!)

Matrix multiplication is not backward stable

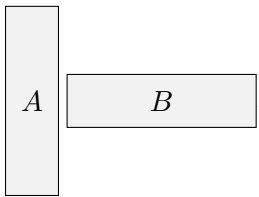
Shock! It is not always true that $fl(AB)$ equal to $(A + \Delta A)(B + \Delta B)$ for small $\Delta A, \Delta B$

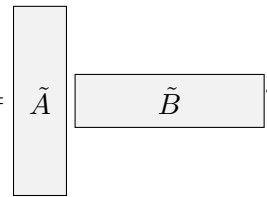
- Vec-vec mult. backward stable: $fl(y^T x) = (y + \Delta y)(x + \Delta x)$; in fact $fl(y^T x) = (y + \Delta y)x$.
- Hence mat-vec also backward stable: $fl(Ax) = (A + \Delta A)x$.
- Still mat-mat is not backward stable.

Matrix multiplication is not backward stable

Shock! It is not always true that $fl(AB)$ equal to $(A + \Delta A)(B + \Delta B)$ for small $\Delta A, \Delta B$

- Vec-vec mult. backward stable: $fl(y^T x) = (y + \Delta y)(x + \Delta x)$; in fact $fl(y^T x) = (y + \Delta y)x$.
- Hence mat-vec also backward stable: $fl(Ax) = (A + \Delta A)x$.
- Still mat-mat is not backward stable.

$AB =$ 

. $fl(AB) = AB + \epsilon =$ 

??

with $\tilde{A} = A + \epsilon\|A\|$, $\tilde{B} = B + \epsilon\|B\|$? No—e.g., $fl(AB)$ is usually not low rank

Matrix multiplication is not backward stable

Shock! It is not always true that $fl(AB)$ equal to $(A + \Delta A)(B + \Delta B)$ for small $\Delta A, \Delta B$

- Vec-vec mult. backward stable: $fl(y^T x) = (y + \Delta y)(x + \Delta x)$; in fact $fl(y^T x) = (y + \Delta y)x$.
- Hence mat-vec also backward stable: $fl(Ax) = (A + \Delta A)x$.
- Still mat-mat is not backward stable.

What **is** true: $\|fl(AB) - AB\| \leq \epsilon \|A\| \|B\|$, so $\|fl(AB) - AB\|/\|AB\| \leq \epsilon \min(\kappa_2(A), \kappa_2(B))$.

- Great when A or B orthogonal (or square well-conditioned): say if $A = Q$ orthogonal,

$$\|fl(QB) - QB\| \leq \epsilon \|B\|,$$

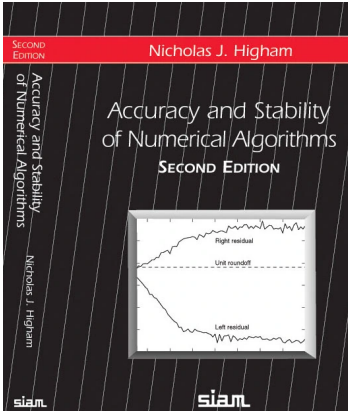
so $fl(QB) = QB + \epsilon \|B\|$, hence $fl(QB) = Q(B + \Delta B)$ where $\Delta B = Q^T \epsilon \|B\|$

orthogonal multiplication is backward stable

Classical/important results in NLA stability

- Orthogonal linear algebra is backward stable backward stable
- Least-squares solved via Householder QR is backward stable (not if normal equations solved)
- Underdetermined system via Householder QR is backward stable
- Cholesky is backward stable
- Triangular solve is backward stable
- LU with partial pivots is usually (not always) backward stable
- $\kappa_2(A) = \|A\|_2\|A^{-1}\|$ is prevalent
- Matrix multiplication AB is **not** backward stable
- ...

all in [Higham ASNA 2002]



(Lack of) stability of randomized NLA algorithms

(Backward) stability usually not discussed when randomized NLA alg introduced

But stability is crucial! In fact some existing algorithms are unstable.

(Lack of) stability of randomized NLA algorithms

(Backward) stability usually not discussed when randomized NLA alg introduced
But stability is crucial! In fact some existing algorithms are unstable.

Remainder: stability of **randomized** NLA algorithms

We'll touch on

I Low-rank approximation (RandSVD, generalized Nyström, Nyström, CUR)
II Sketch-to-precondition $\min_x \|Ax - b\|_2$ and $Ax = b$

Numerical stability of RandSVD

1. Compute sketch $Y = A\Omega$,
2. $A \approx QQ^T A$, where $Y = QR$.

Stability: $\hat{Q}\hat{Q}^T A \approx QQ^T A$ with computed $\hat{Q}$?

- Orthogonal linear algebra? Almost... (orthonormal, not orthogonal).
- Not trivial as $\hat{Q} - Q$ not small if $\kappa_2(A\Omega) \gg 1$ (usually true)

Numerical stability of RandSVD

1. Compute sketch $Y = A\Omega$,
2. $A \approx QQ^TA$, where $Y = QR$.

Stability: $\hat{Q}\hat{Q}^TA \approx QQ^TA$ with computed $\hat{Q}$?

- Orthogonal linear algebra? Almost... (orthonormal, not orthogonal).
- Not trivial as $\hat{Q} - Q$ not small if $\kappa_2(A\Omega) \gg 1$ (usually true)
- Nonetheless, [Connolly-Higham-Pranesh 22]

$$\|QQ^TA - \hat{Q}\hat{Q}^TA\|_F \leq \|A - QQ^TA\|_F + C_{m,n}u\|A\|_F$$

$C_{m,n}$: low-degree polynomial in m, n. **Satisfactory** result.

Proof sketch:

By classical stability of QR, $Y + \Delta Y = \hat{Q}\hat{R}$, $\|\Delta Y\| = O(u\|Y\|) = O(u\|A\|)$.

$$QQ^TA - \hat{Q}\hat{Q}^TA = (I - \hat{Q}\hat{Q}^T)Y = (I - \hat{Q}\hat{Q}^T)\Delta Y$$

so $\|QQ^TA - \hat{Q}\hat{Q}^TA\| \leq \|\Delta Y\| = O(u\|A\|)$.

Generalized Nyström

Stabilized Generalized Nyström (GN) for $A \in \mathbb{R}^{m\times n}$:

$$A \approx AX(Y^TAX)^\dagger Y^TA = \boxed{AX}\,\boxed{(Y^TAX)^\dagger}\,\boxed{Y^TA}$$

- For sketches $X \in \mathbb{R}^{n\times r}, Y \in \mathbb{R}^{m\times(r+\ell)}$, $\ell = cr$ (we choose $c = 0.5$)
- Near-optimal accuracy, comparable to RandSVD: for $\hat{r} = 0.7r$ (say),

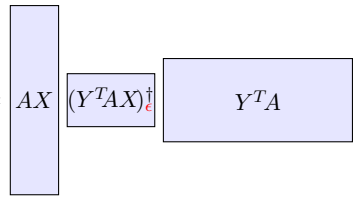
$$\mathbb{E}[\|A - AX(Y^TAX)_\epsilon^\dagger Y^TA\|_F] = O(1)\|A - A_{\hat{r}}\|_F$$

  - $(Y^TAX)_\epsilon^\dagger$ is alarming—ill-conditioned (pseudo)inverse!
  - Poor implementation catastrophic

Generalized Nyström

Stabilized Generalized Nyström (GN) for $A \in \mathbb{R}^{m\times n}$:

$$A \approx AX(Y^TAX)^\dagger_\epsilon Y^TA =$$



- For sketches $X \in \mathbb{R}^{n\times r}, Y \in \mathbb{R}^{m\times(r+\ell)}$, $\ell = cr$ (we choose $c = 0.5$)
- Near-optimal accuracy, comparable to RandSVD: for $\hat{r} = 0.7r$ (say),

$$\mathbb{E}[\|A - AX(Y^TAX)^\dagger_\epsilon Y^TA\|_F] = O(1)\|A - A_{\hat{r}}\|_F$$

  - $(Y^TAX)^\dagger_\epsilon$ is alarming—ill-conditioned (pseudo)inverse!
  - Poor implementation catastrophic
  - Stable with ϵ-pseudoinverse $(U\Sigma V^T)^\dagger_\epsilon = V\Sigma^\dagger_\epsilon U^T$, **if carefully implemened**

Stability analysis sketch: $fl(\hat{A}) = \hat{A}_r + \epsilon$ [N. 20]

$\hat{A} = (AX(Y^TAX)^\dagger_\epsilon)Y^TA$. Each row of $AX(Y^TAX)^\dagger_\epsilon$ is underdetermined linear system, solve via SVD or (rank-revealing) QR.
Define $s_i^T = [AX(Y^TAX)^\dagger_\epsilon]_i$, ith row

$$s_i = ((Y^TAX)^T)^\dagger_\epsilon [AX]_i^T = (X^TA^TY)^\dagger_\epsilon [AX]_i^T =: M_\epsilon^\dagger [AX]_i^T.$$

Computed version satisfies, by [ASNA Ch. 21] ($\hat{U}$: computed Range(M))

$$\hat{s}_i = (\hat{U}^T M + \epsilon)^\dagger (\hat{U}^T [AX]_i^T + \epsilon) = (M + \epsilon_i)^\dagger_\epsilon ([AX]_i^T + \epsilon)_\epsilon.$$

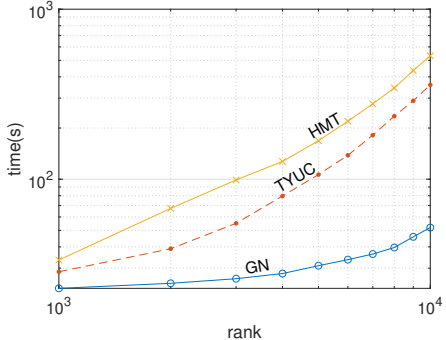
Thus

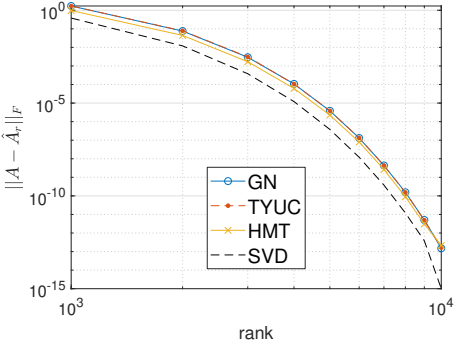
$$\begin{aligned}
[fl(AX(Y^TAX)^\dagger_\epsilon Y^TA)]_i &= fl([AX + \epsilon]_i (Y^TAX + \epsilon_i)^\dagger_\epsilon Y^TA) \\
&= [AX]_i (Y^T\tilde{A}X)^\dagger_\epsilon Y^TA + \epsilon \|[AX]_i (Y^T\tilde{A}X)^\dagger_\epsilon\| \|Y^TA\| \\
&= [AX]_i (Y^T\tilde{A}X)^\dagger_\epsilon Y^TA + \epsilon = [\hat{A}_r]_i + \epsilon
\end{aligned}$$

Row-wise stability follows from $\|AX(Y^TAX)^\dagger\| = O(1)$, $\quad \|AX(Y^T\widetilde{A}X)^\dagger_\epsilon\| = O(1)$ (shown separately).

Experiments: dense matrix

Dense 50000×50000 matrix w/ geom. decaying σ_i





HMT=RandSVD: Halko-Martinsson-Tropp 11, TYUC: Tropp-Yurtsever-Udell-Cevher 17

- GN and TYUC have same accuracy (as they should)
- GN faster, up to ≈ 10x

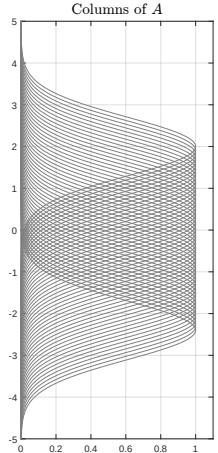
Nyström for $A \succeq 0$

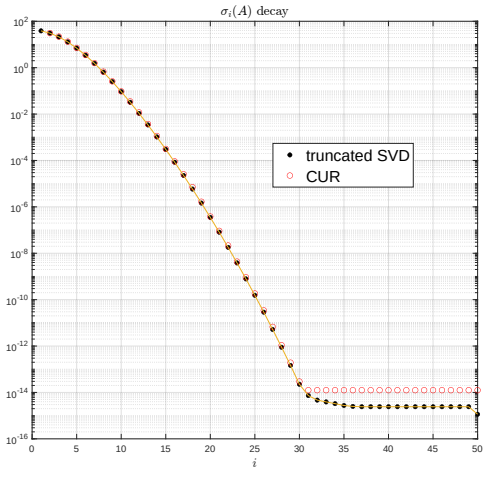
$$A \approx A(:, I)A(I, I)^{-1}A(I, :)$$

- Again, ill-conditioned inverse $A(I, I)^{-1}$ is alarming
- Nyström $=$ Cholesky (in exact arithmetic), and Cholesky is stable.
  But $\not\Longrightarrow$ Nyström is; implementation matters
- One solution is to shift [Tropp-Yurtsever-Udell-Cevher 17], but more expensive and slightly changes approx (see also [Carson-Daužickaite 24])
- Stable provided [Bucci-N.-Park 25]
  - Good (local maxvol) indices I are chosen,
  - Regularized $A(I, I)_{\epsilon}^{-1}$ and carefully implemented

CUR stability is open. Repost: example with $A_{i,j} = \exp(-|x_i - \mu_j|^2/2)$

x_i: 500 equispaced pts in $[-5, 5]$, μ_j: equispaced in $[-2.5, 2]$

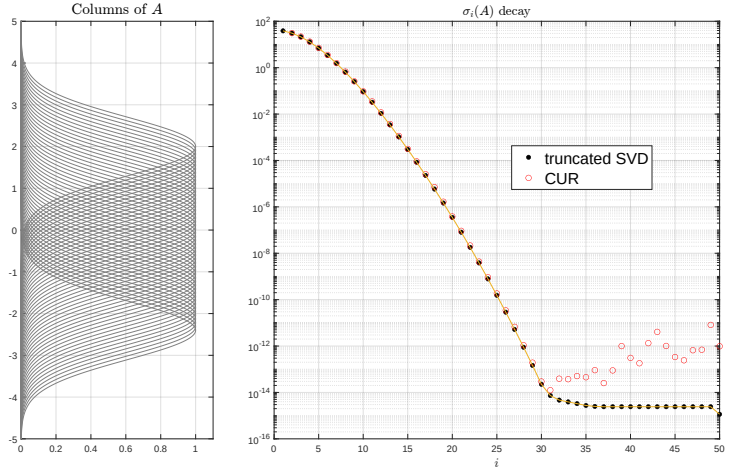




I was the supressing the CUR rank after 30... if I don't,

CUR stability is open. Repost: example with $A_{i,j} = \exp(-|x_i - \mu_j|^2/2)$

x_i: 500 equispaced pts in $[-5, 5]$, μ_j: equispaced in $[-2.5, 2]$



I was the supressing the CUR rank after 30... if I don't,

CUR stability analysis is incomplete. Oversampling $|I| > |J|$ may help [Park-N. 25]

Sketch-to-precondition for least-squares: Blendenpik

[R lin-Tygert 08, Avron-Maymounkov-Toledo 2010]

$$\min_x \|Ax - b\|_2, \qquad A \in \mathbb{R}^{m \times n},\ m \gg n$$

- Traditional method: normal eqn $x = (A^T A)^{-1} A^T b$ or $A = QR, x = R^{-1}(Q^T b)$, both $O(mn^2)$ cost
- Randomized: generate random $G \in \mathbb{R}^{4n \times m}$, and $G \cdot A = \hat{Q}\hat{R}$

  Then solve $\min_y \|(A\hat{R}^{-1})y - b\|_2$ via LSQR (=CG on $A^T Ax = A^T b$)
  - Successful because $A\hat{R}^{-1}$ is well-conditioned

Explaining Blendenpik via Marchenko-Pastur

Claim: $A\hat{R}^{-1}$ is well-conditioned with G A $= \hat{Q}$ $\hat{R}$ (QR)

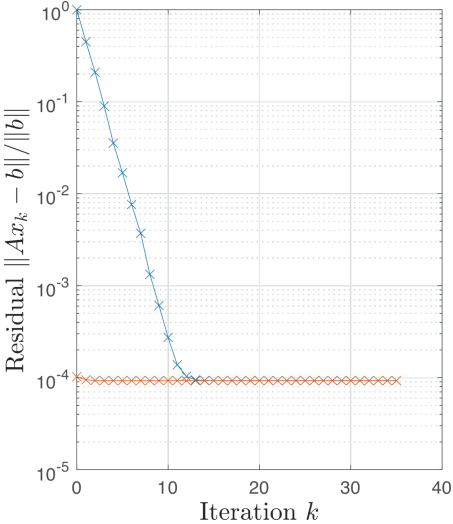
Show this for $G \in \mathbb{R}^{4n \times m}$ Gaussian:

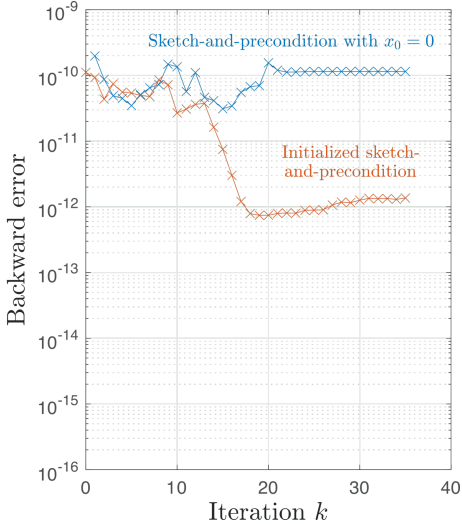
Proof: Let $A = QR$. Then $GA = (GQ)R =: \tilde{G}R$

- $\tilde{G}$ is $4n \times n$ rectangular Gaussian, hence well-cond
- So by M-P, $\kappa_2(\tilde{R}^{-1}) = O(1)$ where $\tilde{G} = \tilde{Q}\tilde{R}$ is QR
- Thus $\tilde{G}R = (\tilde{Q}\tilde{R})R = \tilde{Q}(\tilde{R}R) = \tilde{Q}\hat{R}$, so $\hat{R}^{-1} = R^{-1}\tilde{R}^{-1}$
- Hence $A\hat{R}^{-1} = Q\tilde{R}^{-1}$, $\kappa_2(A\hat{R}^{-1}) = \kappa_2(\tilde{R}^{-1}) = O(1)$

Blendenpik is not stable

[Meier-N.-Townsend-Webb 24, Epperly-Meier-N. 25, slides from Meier]





Sketch-and-solve initialization is helpful, but not perfect

Blendenpik with iterative refinement

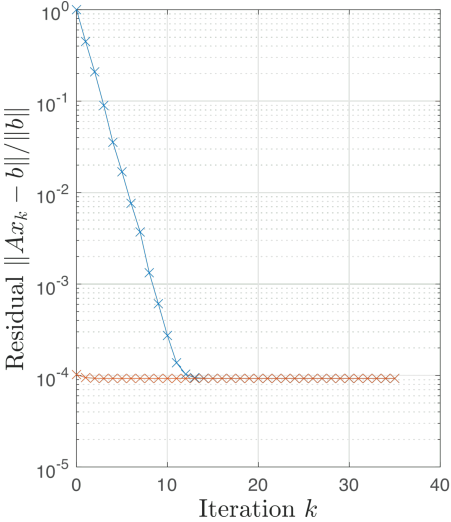
SPIR: Sketch-and-Precondition with Iterative Refinement [Epperly-Meier-N. 25]

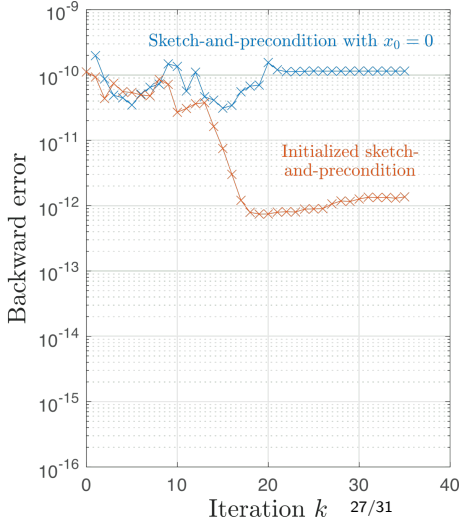
1. Sketch-and-solve initialization
   - 1.1 Draw a random matrix $S \in \mathbb{R}^{s \times m}$, $n < s \ll m$
   - 1.2 Compute SA, Sb, and $SA = QR$
   - 1.3 Set $x_0 = R^{-1}Q^T Sb = \text{argmin}_x \|S(Ax - b)\|_2$
2. Sketch-to-precondition with initial guess x_0
   - 2.1 Compute residual $r_0 = b - Ax_0$
   - 2.2 Compute $\delta x_0 = \text{argmin}_{\delta x} \|A\delta x - r_0\|_2$ with sketch-to-precondition
   - 2.3 Update $x_1 = x_0 + \delta x_0$
3. Sketch-to-precondition with initial guess x_1
   - 3.1 Compute $r_1 = b - Ax_1$
   - 3.2 Compute $\delta x_1 = \text{argmin}_{\delta x} \|A\delta x - r_1\|_2$ with sketch-to-precondition
   - 3.3 Update $x_2 = x_1 + \delta x_1$

Blendenpik is not stable; iterative refinement fixes it!

SPIR: iterative refinement

Sketch-and-apply: compute AR^{-1} explicitly (expensive)

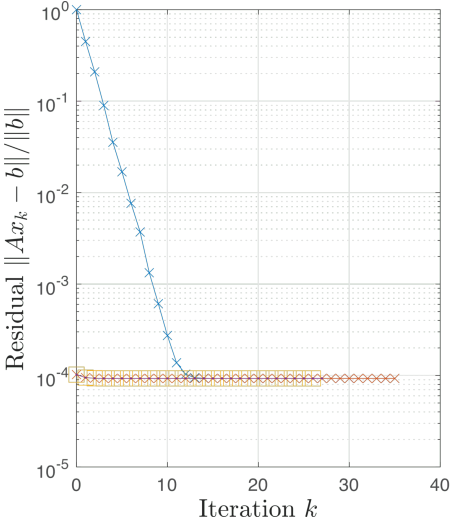


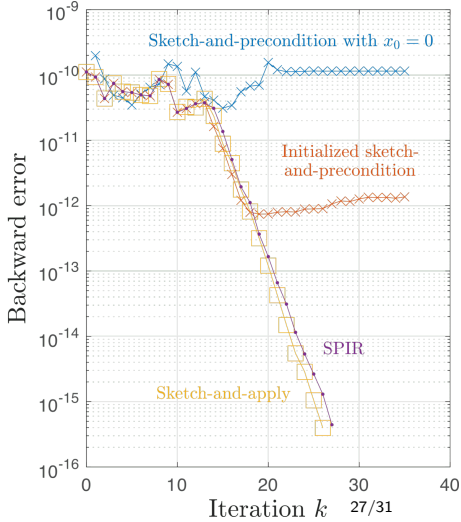


Blendenpik is not stable; iterative refinement fixes it!

SPIR: iterative refinement

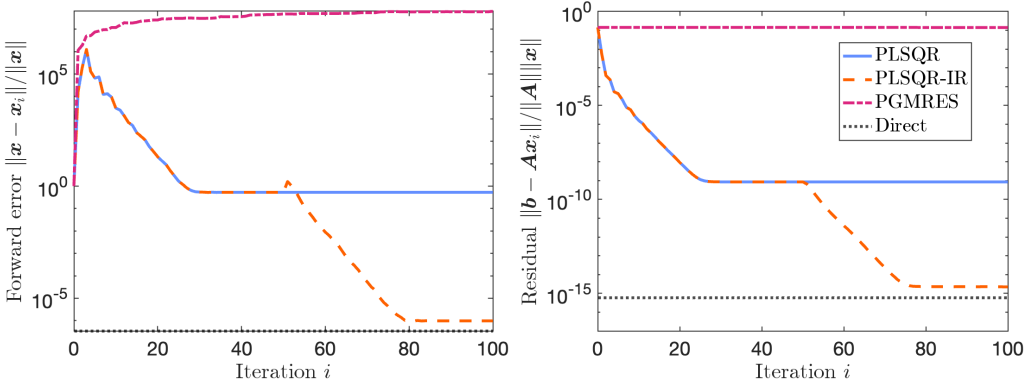
Sketch-and-apply: compute AR^{-1} explicitly (expensive)





Preconditioned LSQR for $Ax = b$ [Epperly-Greenbaum-N. 25]

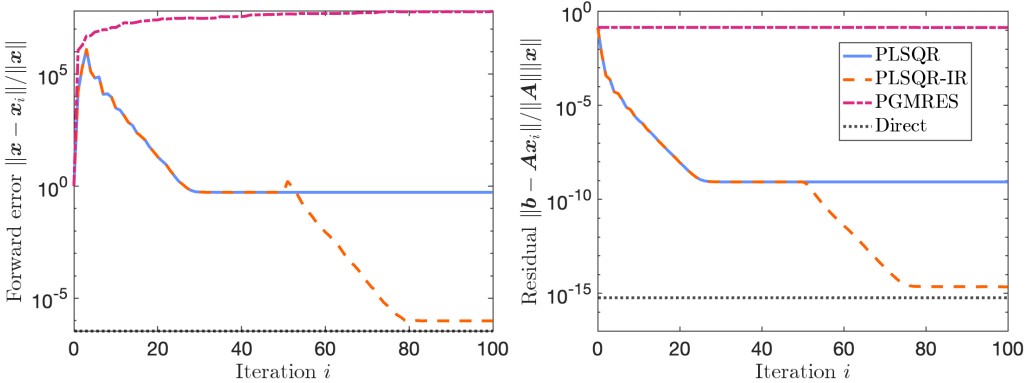
Suppose preconditioner P s.t. $\kappa_2(AP) = O(1)$ given.



- Preconditioned LSQR benefits from iterative refinement, like $\min_x \|Ax - b\|_2$
- Finding such P (cluster σ_i) not easy, but 'easier' than classic goal to cluster λ_i
  - with $\lambda_i \to \lambda_*$, $AP \to \lambda_* I$ (DOF:)
  - $\sigma_i \to \sigma_*$ implies $AP = \sigma_* Q$ for orthonormal Q (DOF:)

Preconditioned LSQR for $Ax = b$ [Epperly-Greenbaum-N. 25]

Suppose preconditioner P s.t. $\kappa_2(AP) = O(1)$ given.



- Preconditioned LSQR benefits from iterative refinement, like $\min_x \|Ax - b\|_2$
- Finding such P (cluster σ_i) not easy, but 'easier' than classic goal to cluster λ_i
  - with $\lambda_i \to \lambda_*$, $AP \to \lambda_* I$ (DOF: 1)
  - $\sigma_i \to \sigma_*$ implies $AP = \sigma_* Q$ for orthonormal Q (DOF: $\frac{n(n+1)}{2}$)

Convergence of iterative methods for $Ax = b$

- CG for $A \succ 0$: $O(n^2\sqrt{\kappa_2(A)}\log(\frac{1}{\epsilon}))$ for ϵ error in A-norm $\|x_* - \hat{x}\|_A / \|x_*\|_A \leq \epsilon$
- GMRES for diagonalizable $A = X\Lambda X^{-1}$: k iterations gives residual

$$\|Ax_k - b\|_2 / \|b\|_2 \leq \kappa_2(X) \min_{p \in \mathcal{P}_k, p(0)=1} \max_{z \in \lambda(A)} |p(z)|.$$

- Recent work [Dereziński-Epperly-Meyer 26] shows $\Omega(\kappa_2(A)\log(1/\epsilon))$ matvecs needed for ϵ guarantee

Convergence of iterative methods for $Ax = b$

- CG for $A \succ 0$: $O(n^2\sqrt{\kappa_2(A)}\log(\frac{1}{\epsilon}))$ for ϵ error in A-norm $\|x_* - \hat{x}\|_A / \|x_*\|_A \leq \epsilon$
- GMRES for diagonalizable $A = X\Lambda X^{-1}$: k iterations gives residual

$$\|Ax_k - b\|_2 / \|b\|_2 \leq \kappa_2(X) \min_{p \in \mathcal{P}_k, p(0)=1} \max_{z \in \lambda(A)} |p(z)|.$$

- Recent work [Dereziński-Epperly-Meyer 26] shows $\Omega(\kappa_2(A)\log(1/\epsilon))$ matvecs needed for ϵ guarantee

Convergence of backward error in iterative methods? [Derezinski-N.-Rebrova 26]

Convergence of iterative methods for $Ax = b$

- CG for $A \succ 0$: $O(n^2\sqrt{\kappa_2(A)}\log(\frac{1}{\epsilon}))$ for ϵ error in A-norm $\|x_* - \hat{x}\|_A / \|x_*\|_A \leq \epsilon$
- GMRES for diagonalizable $A = X\Lambda X^{-1}$: k iterations gives residual

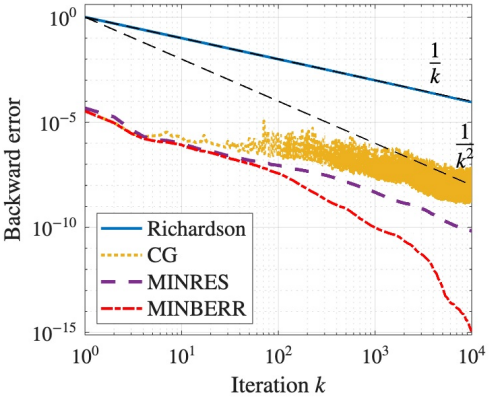
$$\|Ax_k - b\|_2 / \|b\|_2 \leq \kappa_2(X) \min_{p \in \mathcal{P}_k, p(0)=1} \max_{z \in \lambda(A)} |p(z)|.$$

- Recent work [Dereziński-Epperly-Meyer 26] shows $\Omega(\kappa_2(A)\log(1/\epsilon))$ matvecs needed for ϵ guarantee

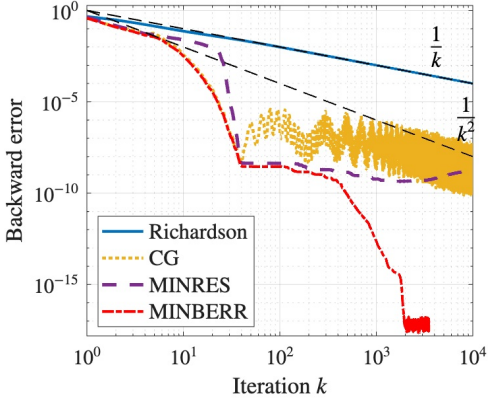
Convergence of backward error in iterative methods? [Derezinski-N.-Rebrova 26]

- $A \succeq 0$: $\kappa(A)$-independent, **universal convergence!**
  - Richardson iteration has complexity $O(\frac{n^2}{\epsilon})$
  - Optimal Krylov (Minimal Backward ERRor: MINBERR): $O(\frac{n^2}{\sqrt{\epsilon}})$
- Nonsymmetric A: $O(\frac{n^2 \log n}{\epsilon})$ via MINBERR on $A^\top A x = A^\top b$
  (uses **randomized perturbation** $A \leftarrow A + \Delta A$)

MINBERR Experiments, matrices from SuiteSparse



$n = 16146$ Aerospace model



$n = 19779$ stability analysis

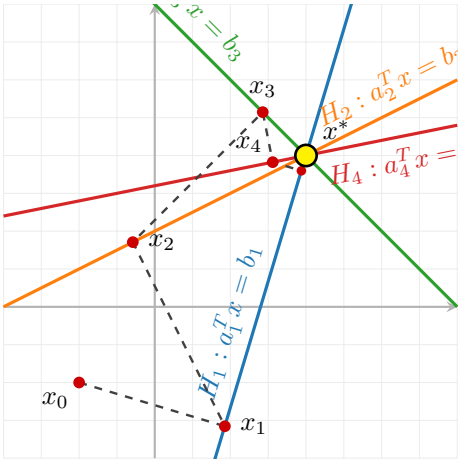
Summary on stability

- Most classical NLA algorithms are backward stable (but not all)
  - **Orthogonal linear algebra** is stable
- Most low-rank methods can be made stable with a **careful implementation**
- Some (randomized) iterative methods aren't stable. **Iterative refinement** can help!
- Classical stability/perturbation analysis continue to be crucial

In the remainder, we could discuss:

- MINBERR in more detail
- Kaczmarz methods
- Matrix learning from matvecs
- Elementwise (max-norm) approximation
- Sketched GMRES and its stability, sketched Rayleigh-Ritz
- Marchenko-Pastur: Fundamental Theorem of Computational Mathematics

Kaczmarz method: successive projections onto hyperplanes



Problem Setup: Solve $Ax = b$, where $A \in \mathbb{R}^{m \times n}$ has rows $a_1^T, \ldots, a_m^T$, and $b \in \mathbb{R}^m$.

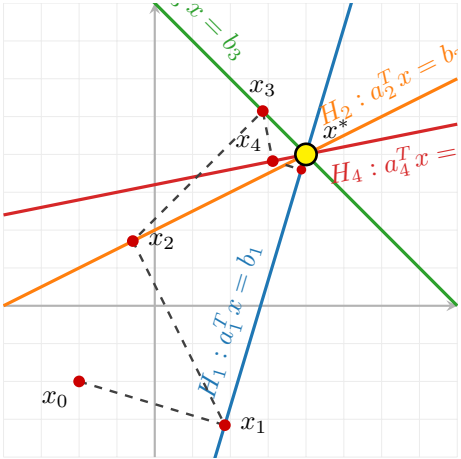
Update Rule: Starting from $x_0 \in \mathbb{R}^n$:

$$x_{k+1} = x_k + \frac{b_{i(k)} - \langle a_{i(k)}, x_k \rangle}{\|a_{i(k)}\|_2^2} a_{i(k)}$$

Geometric Intuition:

- Hyperplanes: $H_i = \{x \in \mathbb{R}^n : \langle a_i, x \rangle = b_i\}$.
- Iteratively pick row $i(k)$ and update soln x_k minimally s.t. $Ax = b$ holds there

Kaczmarz method: successive projections onto hyperplanes



- Classic cyclic ordering $i(k) = (k \bmod m) + 1$: hard to prove convergence
- $O(n)$ operations/iteration (V fast).
- This is SGD (Stochastic Gradient Descent) applied to minimizing $\|Ax - b\|_2^2 = \sum_{i=1}^{N} (a_i x - b_i)^2$: connections to ML

Randomized Kaczmarz [Strohmer–Vershynin 08]

Key Idea: Select row $i(k)$ with probability proportional to row energy:

$$\mathbb{P}(i(k) = i) = \frac{\|a_i\|_2^2}{\|A\|_F^2}$$

Main Theorem (Linear Convergence in Expectation): Let $Ax = b$ be consistent with unique solution x^*. Then:

$$\mathbb{E}\|x_k - x^*\|_2^2 \leq \left(1 - \frac{\sigma_{\min}^2(A)}{\|A\|_F^2}\right)^k \|x_0 - x^*\|_2^2$$

Key Takeaways:

- Guarantees exponential (linear) rate of convergence in expectation.
- Convergence rate depends on scaled condition number $\kappa_F(A) = \frac{\|A\|_F}{\sigma_{\min}(A)}$.

3. Proof Sketch (Strohmer–Vershynin)

1. Orthogonal Error Decomposition: Let error $e_k = x_k - x^*$. Since $b_i = \langle a_i, x^* \rangle$, update step gives $e_{k+1} = e_k - \frac{\langle a_i, e_k \rangle}{\|a_i\|_2^2} a_i$. By Pythagorean theorem:

$$\|e_{k+1}\|_2^2 = \|e_k\|_2^2 - \frac{|\langle a_i, e_k \rangle|^2}{\|a_i\|_2^2}$$

2. Conditional Expectation Given x_k: Summing over choice i with probability $p_i = \frac{\|a_i\|_2^2}{\|A\|_F^2}$:

$$\mathbb{E}_k \|e_{k+1}\|_2^2 = \|e_k\|_2^2 - \sum_{i=1}^{m} \frac{\|a_i\|_2^2}{\|A\|_F^2} \frac{|\langle a_i, e_k \rangle|^2}{\|a_i\|_2^2} = \|e_k\|_2^2 - \frac{\|Ae_k\|_2^2}{\|A\|_F^2}$$

3. Singular Value Lower Bound: Since $\|Ae_k\|_2^2 \geq \sigma_{\min}^2(A) \|e_k\|_2^2$:

$$\mathbb{E}_k \|e_{k+1}\|_2^2 \leq \left(1 - \frac{\sigma_{\min}^2(A)}{\|A\|_F^2}\right) \|e_k\|_2^2$$

Iterating expectation yields the final result. □

Kaczmarz Numerical instability & Stabilization by refinement

[Dereziński, Epperly, Needell, Xue, 2026]

1. **The Finite-Precision Surprise (Lack of Forward Stability):**
   - Standard Randomized Kaczmarz (RK) **not forward stable**
   - Direct solvers achieve forward error $\approx \|x\| \, \tilde{\kappa}(A)u$, where $\tilde{\kappa}(A) = \|A\|_F \|A^{-1}\|$ is the Demmel condition number.
   - RK stagnate at much worse error:

$$\mathbb{E}\|\hat{x}_k - x\| \lesssim \|x\| \, \tilde{\kappa}(A)^2 u$$

   - *Intuition:* Each step incurs $O(u)$ roundoff, damped at rate $1 - O(1/\tilde{\kappa}(A)^2)$, accumulating total error $\sum_{t=0}^{\infty} O(u)(1 - O(1/\tilde{\kappa}(A)^2))^t = O(\tilde{\kappa}(A)^2 u)$.

Kaczmarz Numerical instability & Stabilization by refinement

- Run RK to its error floor $\hat{x}$, compute residual $r = b - A\hat{x}$, solve $Ae = r$ in a separate vector e, and update $\hat{x} \leftarrow \hat{x} + e$.
- Executed entirely in **uniform precision** (no double-precision residual needed).
- **Key Result:** Restores **full forward stability** matching direct solvers:

$$\mathbb{E}\|\hat{x}_t - x\| \lesssim \|x\| \, \tilde{\kappa}(A)u$$

- *Insight:* In exact arithmetic, IR is identical to running more Kaczmarz steps; in floating point, explicit residual tracking prevents error accumulation.

Beyond-Krylov Speedups: Kaczmarz++

Kaczmarz++ *(M Dereziński, D Needell, E Rebrova, J Yang 25)*

- **Motivation:** Standard Kaczmarz struggles on ill-conditioned A with spectral outliers.
- **Spectral Capture:** Captures top-k outlying singular values provably faster than Krylov solvers (CG / GMRES).
- **Convergence Guarantee:**

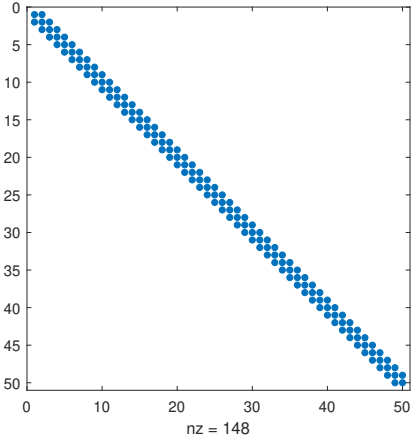
$$\mathbb{E}\|x_t - x^*\|_2^2 \leq 8\left(1 - \frac{k^2}{24m\,\bar{\kappa}_k}\right)^t \|x_0 - x^*\|_2^2$$

  where $\bar{\kappa}_k$ depends only on the remaining spectrum after removing the top k values.

Extensions & Stability

- **CD++:** Adapts Kaczmarz++ to coordinate descent for PSD matrices without inner solvers.
- **Forward Stability:** Fixes error accumulation via randomized iterative refinement.

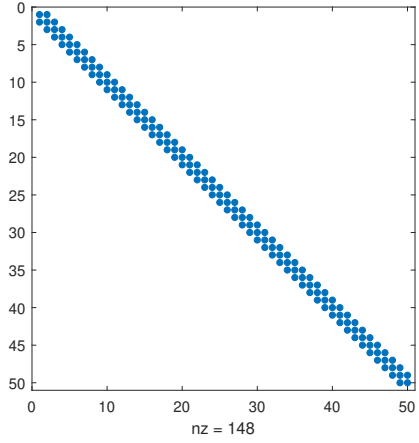
Miscellaneous: matrix learning from matvecs



- $A \in \mathbb{R}^{n \times n}$ is known to be tridiagonal
- We can compute $x^T A$ for any $x \in \mathbb{R}^n$. (e.g. $A = B^3$, B known)

Question: How many vectors x do we need to recover A exactly?

Miscellaneous: matrix learning from matvecs

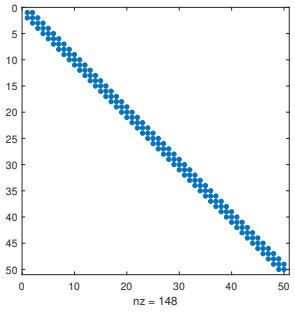


- $A \in \mathbb{R}^{n \times n}$ is known to be tridiagonal
- We can compute $x^T A$ for any $x \in \mathbb{R}^n$. (e.g. $A = B^3$, B known)

Question: How many vectors x do we need to recover A exactly?

Answer: 3

Tridiagonal estimation/recovery



For the ith column of $B = X^T A$,

$$b_i := X^T a_i$$

$$= x_{i-1}a_{i-1,i} + x_i a_{i,i} + x_{i+1}a_{i+1,i}$$

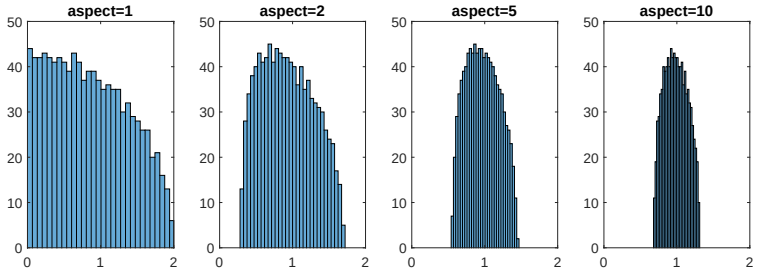
so solve $\min_x \left\| \left[x_{i-1}, x_i, x_{i+1} \right] \begin{bmatrix} a_{i-1,i} \\ a_{i,i} \\ a_{i+1,i} \end{bmatrix} - b_i \right\|$

- Here we can take $X^T = [I_3, I_3, \ldots, I_3]$ [Halikias–Townsend (24)]
- Alternatively, X^T iid Gaussian $X_{ij} \sim N(0,1)$
- Important fact: by Marchenko-Pasture rule, rectangular random matrices are well conditioned, so least-squares give stable results

Marchenko-Pastur rule: Fundamental Theorem of Comp Math

G: random $m \times n$ iid entries, mean 0 variance 1 (e.g. Gaussian)

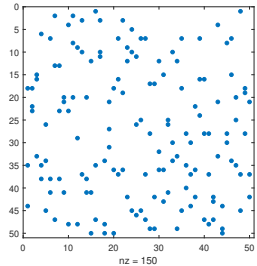
Histogram of singular values $\sigma_i(G)$

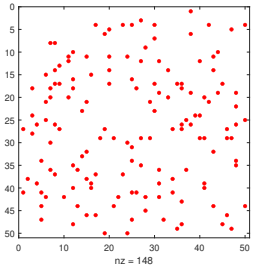


density $\sim \frac{1}{x}\sqrt{(\sqrt{m}+\sqrt{n})-x)(x-(\sqrt{m}-\sqrt{n})}$, support $[\sqrt{m}-\sqrt{n}, \sqrt{m}+\sqrt{n}]$

- $\kappa_2(A) \approx \frac{\sqrt{m}+\sqrt{n}}{\sqrt{m}-\sqrt{n}}$, 'Rectangular random matrix is well-conditioned'
- Related to Johnson-Lindenstrauss Lemma, Compressed sensing Restricted Isometry Property, Oblivious Subspace Embedding, Concentration Inequalities

Pop quiz 2

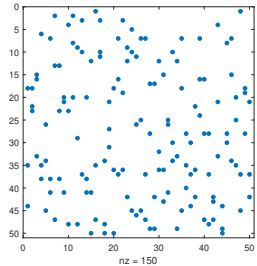


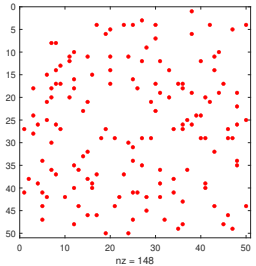


- $A \in \mathbb{R}^{n \times n}$ sparse with ≤ 3 nonzeros per column in **known** location (left), and **unknown** location (right).

Question: How many x do we need to recover A exactly?

Pop quiz 2





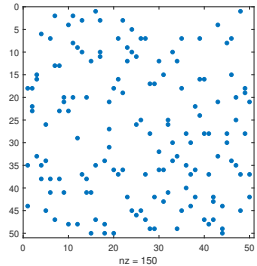
- $A \in \mathbb{R}^{n \times n}$ sparse with ≤ 3 nonzeros per column in **known** location (left), and **unknown** location (right).

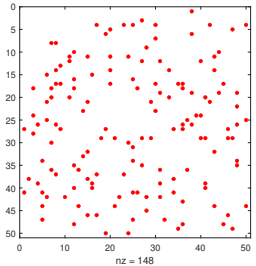
Question: How many x do we need to recover A exactly?

Answer: Left: 3–4 (least-squares as before)

[Amsel–Chen–Halikias–Keles–Musco–Musco arXiv 2024]

Pop quiz 2





- $A \in \mathbb{R}^{n \times n}$ sparse with ≤ 3 nonzeros per column in **known** location (left), and **unknown** location (right).

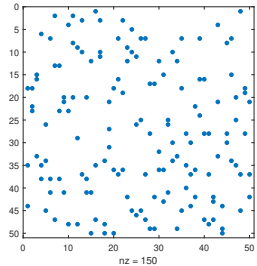
Question: How many x do we need to recover A exactly?

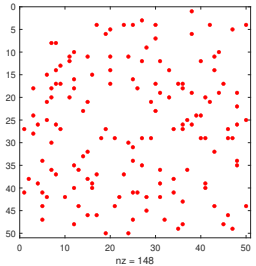
Answer: Left: 3–4 (least-squares as before)

[Amsel–Chen–Halikias–Keles–Musco–Musco arXiv 2024]

Right: $O(\log(n))$ (right; using compressed sensing)

Pop quiz 2





- $A \in \mathbb{R}^{n \times n}$ sparse with ≤ 3 nonzeros per column in **known** location (left), and **unknown** location (right).

Question: How many x do we need to recover A exactly?

Answer: Left: 3–4 (least-squares as before)

[Amsel–Chen–Halikias–Keles–Musco–Musco arXiv 2024]

Right: $O(\log(n))$ (right; using compressed sensing)

Open: What is the class of matrices recoverable from $X^T A$, X Gaussian?

[Levitt-Martinsson arXiv 2022, Boullé-Halikias-Otto-Townsend arXiv 2024 etc]

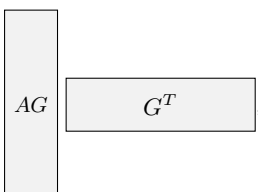
Can we get e.g. eigenvalues directly?

Miscellaneous: Elementwise approximation of a matrix

For **any** $A \in \mathbb{R}^{n \times n}$, there exists rank-$r$ matrix A_r s.t.

$$\|A - A_r\|_{\max} \leq O\left(\frac{\log n}{\sqrt{r}}\right) \|A\|_2$$

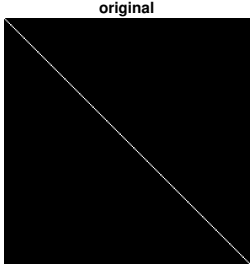
where $\|B\|_{\max} = \max_{i,j} |B_{ij}|$ [Udell-Townsend (2019)]

- Any matrix is approximately low-rank in the **elementwise** sense
- A_r can be taken as $U\Sigma^{1/2} G G^T \Sigma^{1/2} V^*/r$, where $G \in \mathbb{R}^{n \times r}$ Gaussian $G_{ij} \sim N(0,1)$ iid
- Simpler: $A \approx AGG^T =$ 

, where G scaled Gaussian

Low-rank approximation via $A \approx A_r = AGG^T$, G Gaussian

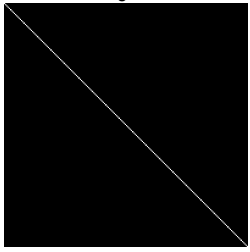
A: identity matrix



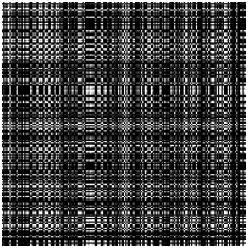
Low-rank approximation via $A \approx A_r = AGG^T$, G Gaussian

A: identity matrix

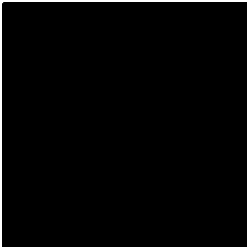
original



elementwise r=1



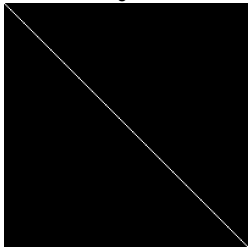
truncated SVD



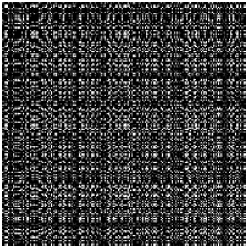
Low-rank approximation via $A \approx A_r = AGG^T$, G Gaussian

A: identity matrix

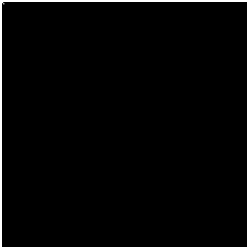
original



elementwise r=2

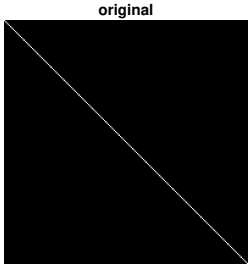


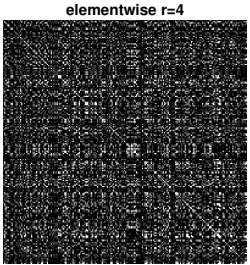
truncated SVD

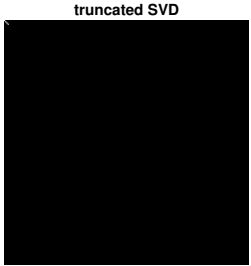


Low-rank approximation via $A \approx A_r = AGG^T$, G Gaussian

A: identity matrix



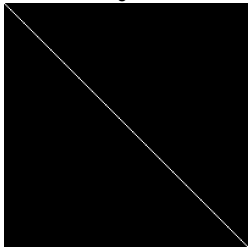




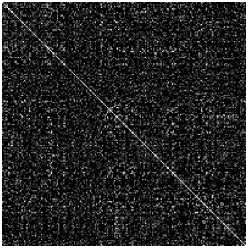
Low-rank approximation via $A \approx A_r = AGG^T$, G Gaussian

A: identity matrix

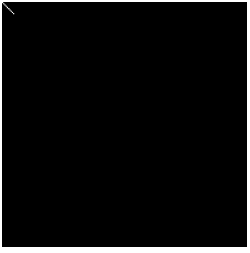
original



elementwise r=10

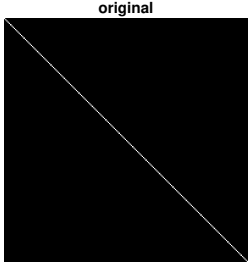


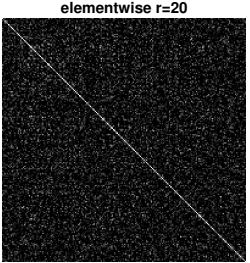
truncated SVD

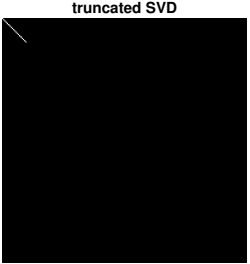


Low-rank approximation via $A \approx A_r = AGG^T$, G Gaussian

A: identity matrix



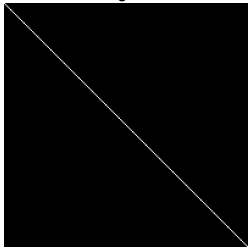




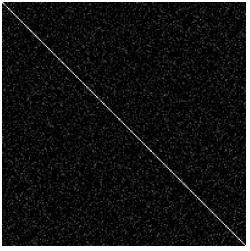
Low-rank approximation via $A \approx A_r = AGG^T$, G Gaussian

A: identity matrix

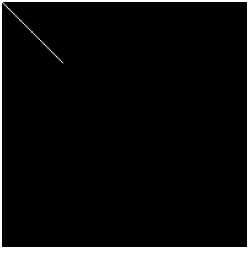
original



elementwise r=50

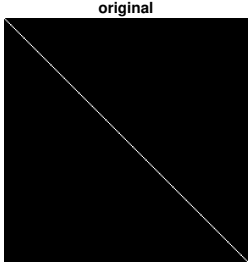


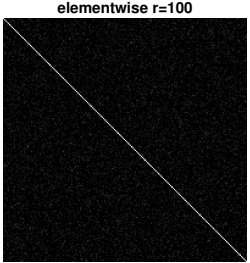
truncated SVD

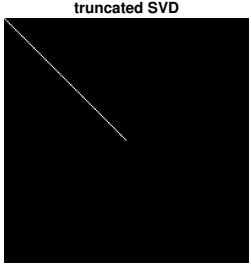


Low-rank approximation via $A \approx A_r = AGG^T$, G Gaussian

A: identity matrix



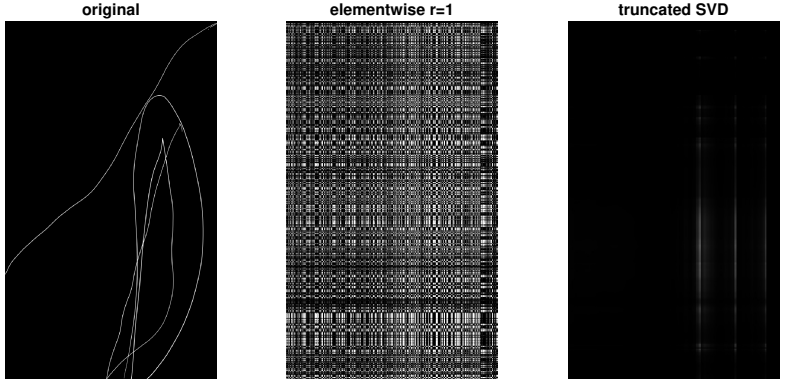




- Not great for high accuracy
- Can be good for low accuracy, detecting heavy hitters etc
- Most trace estimation methods are based on $A \approx AGG^T$:
  $\text{Trace}(A) \approx \text{Trace}(AGG^T) = \text{Trace}(G^T AG) = \sum_{i=1}^{r} g_i^T A g_i$
- Extension to diagonal/matrix estimation in max-norm [Baston-N. 22]
- Open: more applications?

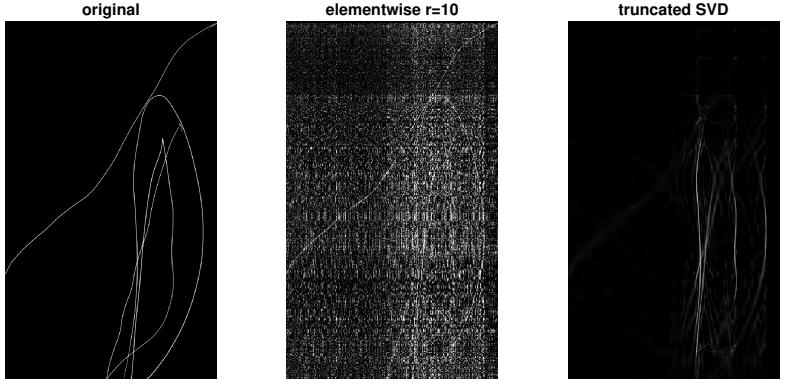
Low-rank approximation of image

[art by E. Nakatsukasa 2020]



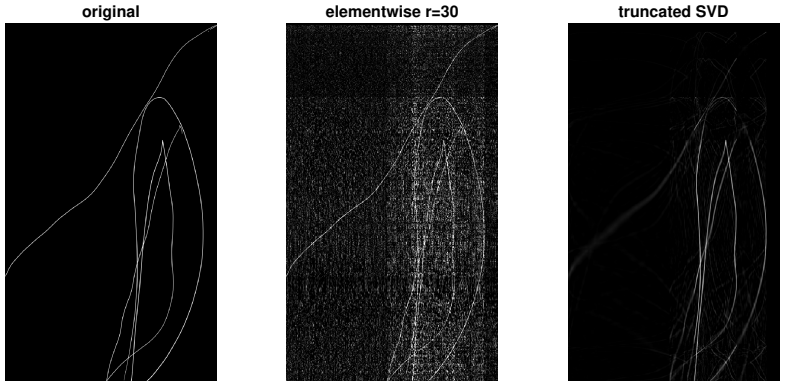
Low-rank approximation of image

[art by E. Nakatsukasa 2020]



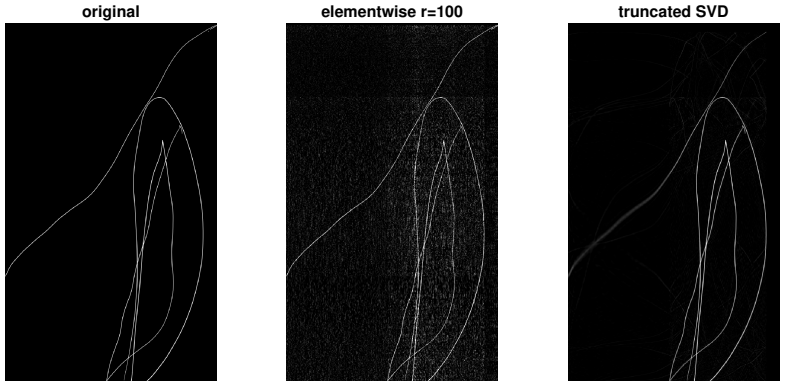
Low-rank approximation of image

[art by E. Nakatsukasa 2020]



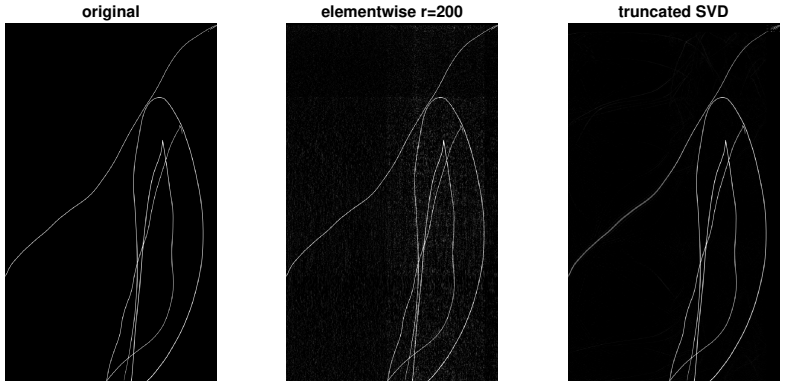
Low-rank approximation of image

[art by E. Nakatsukasa 2020]



Low-rank approximation of image

[art by E. Nakatsukasa 2020]



GMRES for $Ax = b$ review [Saad-Schulz 86]

Minimize residual in Krylov subspace $\mathcal{K}_k(A, b) := \text{span}(b, Ab, \ldots, A^{k-1}b)$

$$x_k = \text{argmin}_{x \in \mathcal{K}_k(A,b)} \|Ax - b\|_2$$

i.e., find solution of form $x_k = P_{k-1}(A)b$, P_{k-1} polynomial deg $\leq k - 1$

GMRES for $Ax = b$ review [Saad-Schulz 86]

Minimize residual in Krylov subspace $\mathcal{K}_k(A, b) := \text{span}(b, Ab, \ldots, A^{k-1}b)$

$$x_k = \text{argmin}_{x \in \mathcal{K}_k(A,b)} \|Ax - b\|_2$$

i.e., find solution of form $x_k = P_{k-1}(A)b$, P_{k-1} polynomial deg $\leq k - 1$

Arnoldi process: finds orthonormal basis Q_k for $\mathcal{K}_k(A, b)$

Set $q_1 = b/\|b\|_2$
For $k = 1, 2, \ldots,$
  set $v = Aq_k$
    for $j = 1, 2, \ldots, k$
      $h_{jk} = q_j^T v$, $v = v - h_{jk} q_j$ % MGS (or CGS2) orthogonalization
    end for
      $h_{k+1,k} = \|v\|_2$, $q_{k+1} = v/h_{k+1,k}$
End for

This gives Arnoldi decomposition $AQ_k = Q_{k+1}\tilde{H}_k$

GMRES for $Ax = b$ review [Saad-Schulz 86]

Minimize residual in Krylov subspace $\mathcal{K}_k(A, b) := \text{span}(b, Ab, \ldots, A^{k-1}b)$

$$x_k = \text{argmin}_{x \in \mathcal{K}_k(A,b)} \|Ax - b\|_2$$

i.e., find solution of form $x_k = P_{k-1}(A)b$, P_{k-1} polynomial deg $\leq k-1$

GMRES: Run Arnoldi $AQ_k = Q_{k+1}\tilde{H}_k$ and write $x_k = Q_k y$, solve

$$\begin{aligned} \min_y \|AQ_k y - b\|_2 &= \min_y \|Q_{k+1}\tilde{H}_k y - b\|_2 \\ &= \min_y \left\| \begin{bmatrix} \tilde{H}_k \\ 0 \end{bmatrix} y - \|b\|_2 e_1 \right\|_2, \quad e_1 = [1, 0, \ldots, 0]^T \end{aligned}$$

- Reduces to Hessenberg least-squares, $O(k^2)$ work
- Overall, k A-mult. + $O(nk^2)$ Arnoldi orthogonalization + $O(k^2)$ Hessenberg solve.
- Orthogonalization $O(nk^2)$ expensive→traditionally, restart GMRES

Does sketching help in GMRES?

- Apparently NOT, as $\tilde{H}_k$ is almost square+full-rank in

$$\min_y \left\| AQ_k \, y - b \right\|_2 = \min_y \left\| \tilde{H}_k \, y - e_1 \right\|_2$$

Does sketching help in GMRES?

- Apparently NOT, as $\tilde{H}_k$ is almost square+full-rank in

$$\min_y \left\| AQ_k \, y - b \right\|_2 = \min_y \left\| \tilde{H}_k \, y - e_1 \right\|_2$$

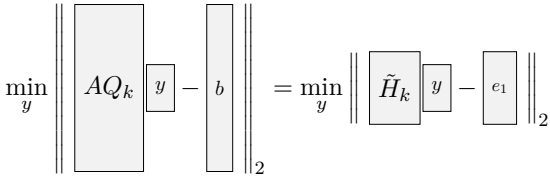
- But Q_k **need not be orthonormal!** Instead, we can find basis B_k s.t. $\mathsf{span}(B_k) = \mathsf{span}(Q_k)$ and solve

$$\min_y \left\| AB_k \, z - b \right\|_2$$

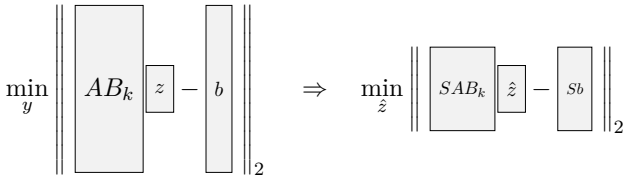
  - $x_k = B_k z$ is still the GMRES solution

Does sketching help in GMRES?

- Apparently NOT, as $\tilde{H}_k$ is almost square+full-rank in



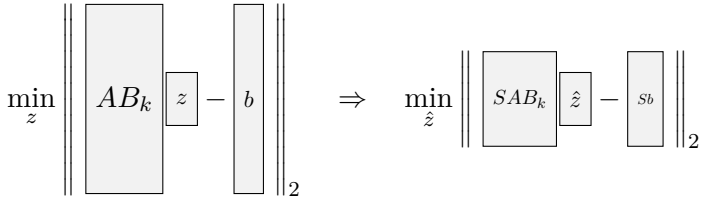
- But Q_k **need not be orthonormal!** Instead, we can find basis B_k s.t. span(B_k) = span(Q_k) and solve



  - $x_k = B_k z$ is still the GMRES solution
  - AB_k is $n \times k$, ripe for sketching! Great if GMRES residual small

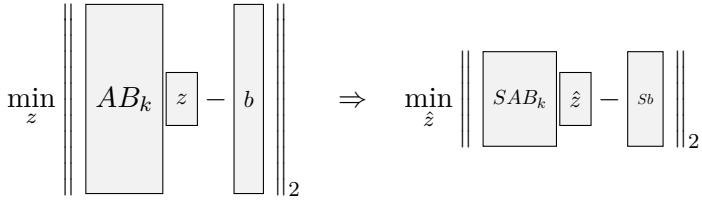
sGMRES (sketched GMRES). Any good?

Non-orthogonal linear algebra



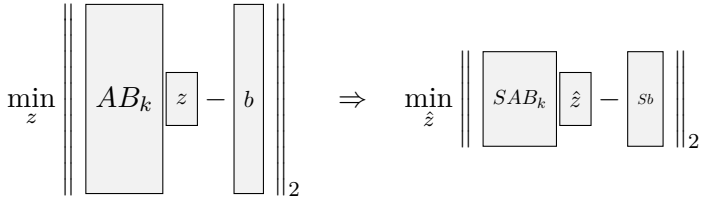
- In GMRES, B_k orthonormal; $O(nk^2)$ cost
- Not necessary! Works fine as long as $\kappa_2(AB_k) < u^{-1} \approx 10^{16}$
- **Why?**

Non-orthogonal linear algebra



- In GMRES, B_k orthonormal; $O(nk^2)$ cost
- Not necessary! Works fine as long as $\kappa_2(AB_k) < u^{-1} \approx 10^{16}$
- **Why?** Because by ASNA Ch. 20, the computed z essentially minimizes $\|AB_k z - b\|_2$, equivalent to GMRES $\|AQ_k y - b\|_2$

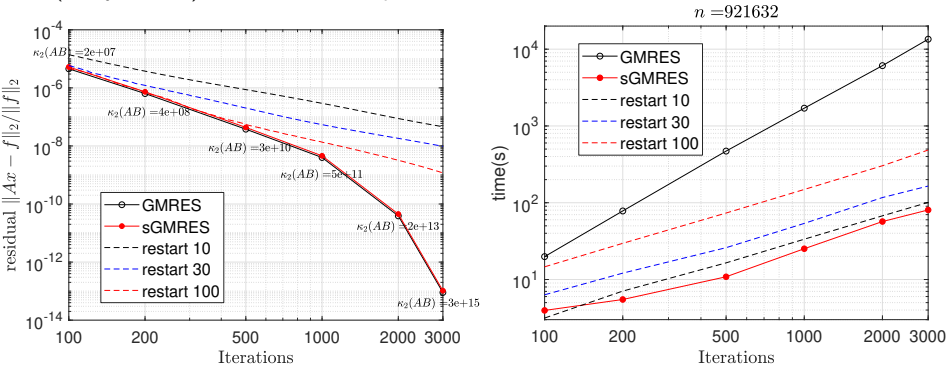
Non-orthogonal linear algebra



- In GMRES, B_k orthonormal; $O(nk^2)$ cost
- Not necessary! Works fine as long as $\kappa_2(AB_k) < u^{-1} \approx 10^{16}$
- **Why?** Because by ASNA Ch. 20, the computed z essentially minimizes $\|AB_k z - b\|_2$, equivalent to GMRES $\|AQ_k y - b\|_2$
- Same for sketching step
- Offers enormous flexibility in choice of basis B_k, eliminating need for Arnoldi orthogonalization
- Forming $[b, Ab, A^2b, \ldots]$ is still bad idea—we suggest truncated/partial orthogonalization; $O(nk)$ cost

Experiments with sGMRES

t2em (nonsymmetric) matrix from SuiteSparse Matrix Collection



- sGMRES $(p = 4)$ achieves almost 100x speedup over GMRES, comparable to GMRES-10 (restarted every 10 iterations)
- Accuracy of sGMRES is nearly identical to GMRES, while B_k and AB_k are full rank [Burke-Carson-Ma 25]

III. Eigenvalue problems: Review of Rayleigh-Ritz

$$Ax = \lambda x$$

- If A modest size $\lesssim 5000$, standard QR alg. is amazing
- For larger problems, **subspace methods**: find subspace $B \in \mathbb{R}^{n \times k}$ that approximately contains desired eigenvector(s), and solve.

III. Eigenvalue problems: Review of Rayleigh-Ritz

$$Ax = \lambda x$$

- If A modest size $\lesssim 5000$, standard QR alg. is amazing
- For larger problems, **subspace methods**: find subspace $B \in \mathbb{R}^{n \times k}$ that approximately contains desired eigenvector(s), and solve.

Classical approach: **Rayleigh-Ritz** (RR); $B = QR$ (if B not orthonormal), and solve small eigenproblem

$$Q^T A Q y = \lambda y.$$

(λ, Qy) is approximate eigenpair (Ritz pair). Cost (at least) $O(nk^2)$

III. Eigenvalue problems: Review of Rayleigh-Ritz

$$Ax = \lambda x$$

- If A modest size $\lesssim 5000$, standard QR alg. is amazing
- For larger problems, **subspace methods**: find subspace $B \in \mathbb{R}^{n \times k}$ that approximately contains desired eigenvector(s), and solve.

Classical approach: **Rayleigh-Ritz** (RR); $B = QR$ (if B not orthonormal), and solve small eigenproblem

$$Q^T A Q y = \lambda y.$$

(λ, Qy) is approximate eigenpair (Ritz pair). Cost (at least) $O(nk^2)$

Can we sketch Rayleigh-Ritz? Doesn't look that way...

Alternative formulation of Rayleigh-Ritz

Key fact: RR is equivalent to solving $My = \lambda y$, where [Parlett 96, Ch. 11]

$$\underset{M \in \mathbb{R}^{k \times k}}{\text{minimize}} \left\| AB - BM \right\|_F ,$$

Alternative formulation of Rayleigh-Ritz

Key fact: RR is equivalent to solving $My = \lambda y$, where [Parlett 96, Ch. 11]

$$\underset{M \in \mathbb{R}^{k \times k}}{\text{minimize}} \left\| AB - BM \right\|_F ,$$

This can be sketched! **sRR** (sketched Rayleigh-Ritz)

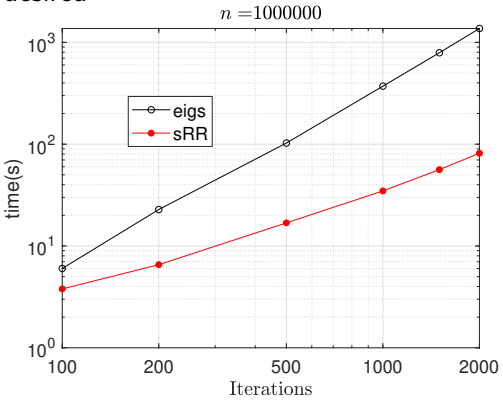
$$\underset{\hat{M} \in \mathbb{R}^{k \times k}}{\text{minimize}} \left\| SAB - SB\hat{M} \right\|_F$$

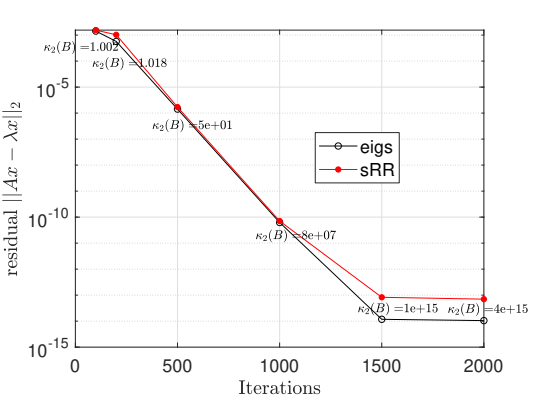
- Small residual $\|ABy - \lambda By\| \Rightarrow$ small $\|S(ABy - \lambda By)\|$
- However, stability analysis is incomplete, or impossible: Stewart's classical example shows RR (without sketching) can fail (but it works almost always; a bit like GEPP).

Experiments: nonsymmetric eigs

B: Krylov (for comparison with eigs)

Nonsymmetric eigenproblem arising in trust-region subproblem, rightmost eigenpair desired





What do direct ('best') methods achieve?

$A = PLU$ or $A = QR$ requiring $O(n^3)$ ~~(or $O(n^\omega)$)~~, gives computed $\hat{x}$ s.t.

- $\hat{x}$ is backward stable $(A+\Delta A)\hat{x} = b$, $\|\Delta A\|/\|A\| = O(u)$
- But this implies NOTHING about forward error $\|x_* - \hat{x}\|/\|x_*\|$; can be as large as $u\kappa_2(A)$

What do direct ('best') methods achieve?

$A = PLU$ or $A = QR$ requiring $O(n^3)$ ~~(or $O(n^\omega)$)~~, gives computed $\hat{x}$ s.t.

- $\hat{x}$ is backward stable $(A+\Delta A)\hat{x} = b$, $\|\Delta A\|/\|A\| = O(u)$
- But this implies NOTHING about forward error $\|x_* - \hat{x}\|/\|x_*\|$; can be as large as $u\kappa_2(A)$

So..

- Requiring ϵ forward error is arguably preposterous
- We should examine convergence also in backward error!

What do direct ('best') methods achieve?

$A = PLU$ or $A = QR$ requiring $O(n^3)$ ~~(or $O(n^\omega)$)~~, gives computed $\hat{x}$ s.t.

- $\hat{x}$ is backward stable $(A+\Delta A)\hat{x} = b$, $\|\Delta A\|/\|A\| = O(u)$
- But this implies NOTHING about forward error $\|x_* - \hat{x}\|/\|x_*\|$; can be as large as $u\kappa_2(A)$

So..

- Requiring ϵ forward error is arguably preposterous
- We should examine convergence also in backward error!

Existing backward error (berr) analysis:

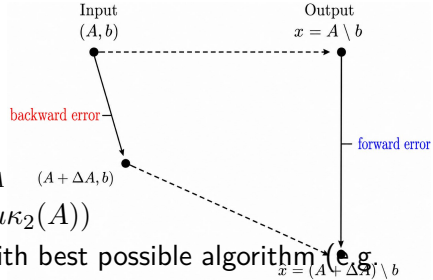
- Mostly for stability analysis of algorithms [Wilkinson 65, Jankowski-Wozniakowski 77, Greenbaum 89, Higham 02 etc etc]
- Surprisingly, no prior work looks at convergence of basic iterative methods wrt berr

Forward vs Backward error $\widehat{x}$

Forward error: $\frac{\|\widehat{x}-x_*\|}{\|x_*\|}$

- '$\widehat{x}$ is a slightly wrong solution'
- includes relative residual, by taking $\|\cdot\| = \|\cdot\|_{A^\top A}$
- unavoidably affected by conditioning $\frac{\|\widehat{x}-x_*\|}{\|x_*\|} \leq O(u\kappa_2(A))$
- impossible to get ϵ forward error for any A, even with best possible algorithm (e.g. $A\backslash b$ doesn't, even for residual)

Forward vs Backward error $\widehat{x}$



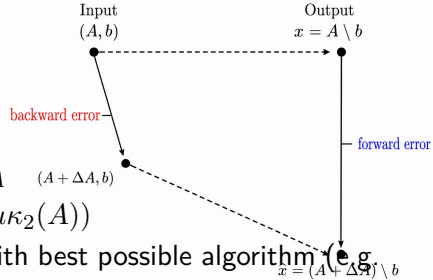
Forward error: $\frac{\|\widehat{x}-x_*\|}{\|x_*\|}$

- '$\widehat{x}$ is a slightly wrong solution'
- includes relative residual, by taking $\|\cdot\| = \|\cdot\|_{A^\top A}$
- unavoidably affected by conditioning $\frac{\|\widehat{x}-x_*\|}{\|x_*\|} \leq O(u\kappa_2(A))$
- impossible to get ϵ forward error for any A, even with best possible algorithm (e.g. $A\backslash b$ doesn't, even for residual)

Backward error $\quad \text{berr}_{A,b}(\widehat{x}) := \min_{\Delta A} \frac{\|\Delta A\|}{\|A\|} \quad \text{s.t.} \quad (A + \Delta A)\widehat{x} = b$

- '$\widehat{x}$ is exact soln of slightly wrong problem'
- gold standard of good numerical algorithms [Higham 2002]
- $\text{berr}_{A,b}(x) = \frac{\|Ax-b\|}{\|A\|\|x\|}$.
- Forward error$\lesssim \kappa \cdot$berr

Forward vs Backward error $\widehat{x}$



Forward error: $\frac{\|\widehat{x}-x_*\|}{\|x_*\|}$

- '$\widehat{x}$ is a slightly wrong solution'
- includes relative residual, by taking $\|\cdot\| = \|\cdot\|_{A^\top A}$
- unavoidably affected by conditioning $\frac{\|\widehat{x}-x_*\|}{\|x_*\|} \leq O(u\kappa_2(A))$
- impossible to get ϵ forward error for any A, even with best possible algorithm (e.g. $A\setminus b$ doesn't, even for residual)

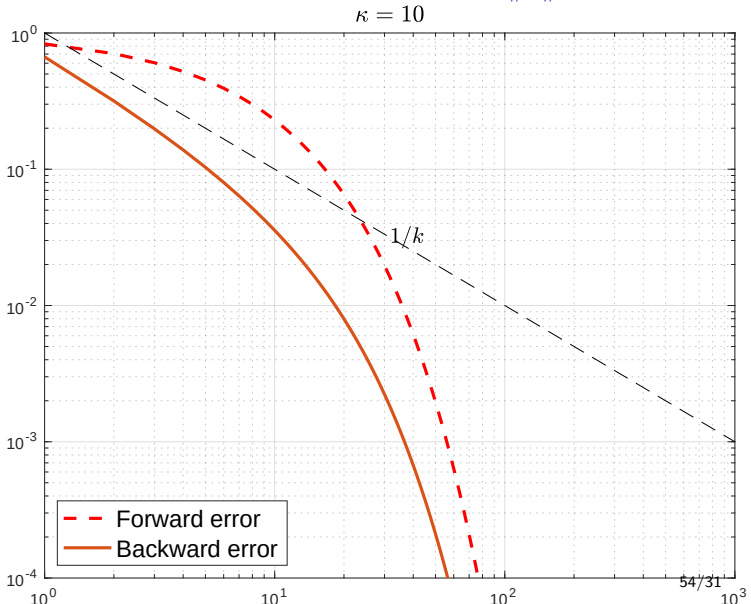
Backward error $\quad \text{berr}_{A,b}(\widehat{x}) := \min_{\Delta A} \frac{\|\Delta A\|}{\|A\|} \quad \text{s.t.} \quad (A+\Delta A)\widehat{x} = b$

- '$\widehat{x}$ is exact soln of slightly wrong problem'
- gold standard of good numerical algorithms [Higham 2002]
- $\text{berr}_{A,b}(x) = \frac{\|Ax-b\|}{\|A\|\|x\|}$.
- Forward error$\lesssim \kappa\cdot$berr

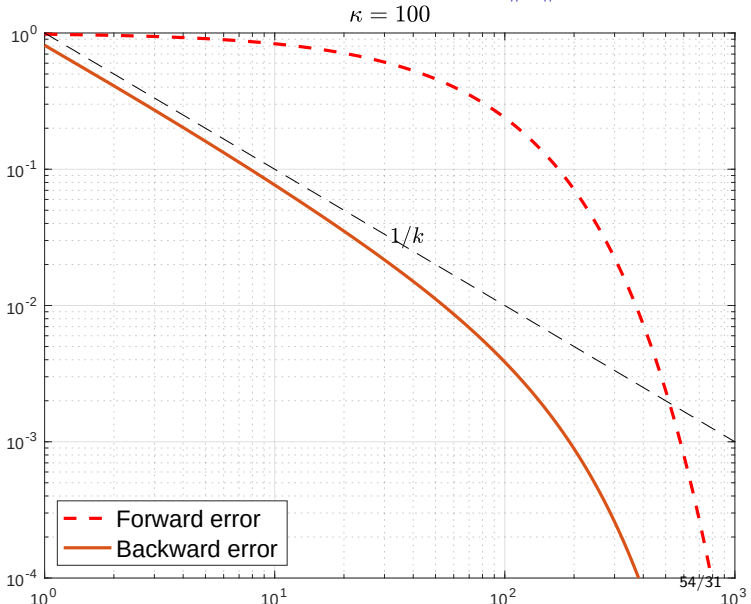
Most convergence analysis: on forward error

This work: convergence analysis of **backward error**

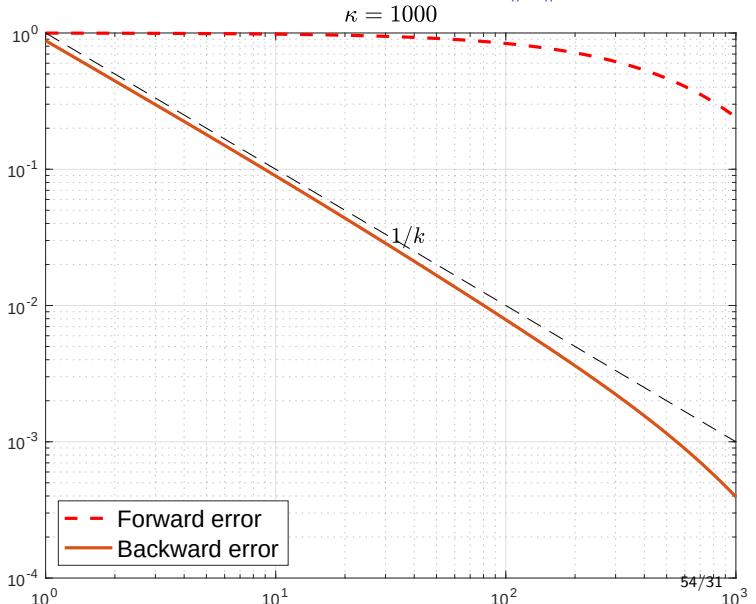
Richardson iteration for $A \succeq 0$: $x_{k+1} = x_k - \frac{1}{\|A\|_2}(Ax_k - b)$.



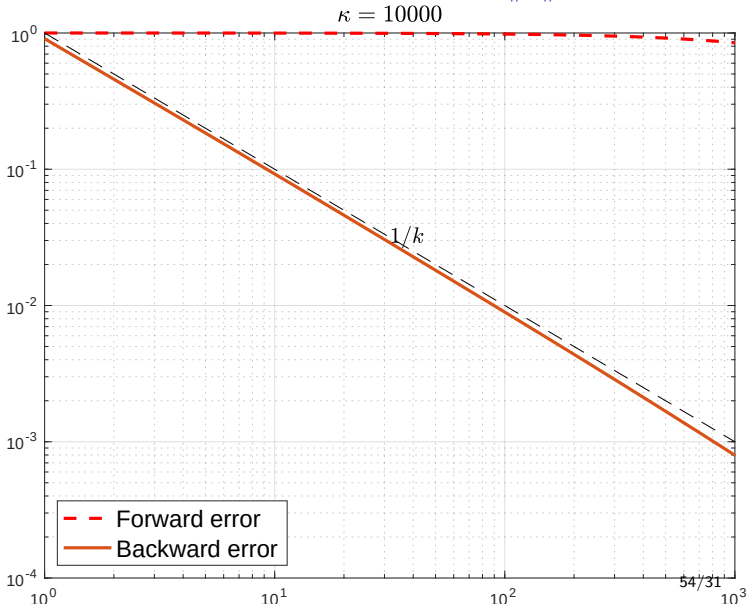
Richardson iteration for $A \succeq 0$: $x_{k+1} = x_k - \frac{1}{\|A\|_2}(Ax_k - b)$.



Richardson iteration for $A \succeq 0$: $x_{k+1} = x_k - \frac{1}{\|A\|_2}(Ax_k - b)$.



Richardson iteration for $A \succeq 0$: $x_{k+1} = x_k - \frac{1}{\|A\|_2}(Ax_k - b)$.



Richardson iteration complexity is $O(n^2/\epsilon)$ (V surprising to me!)

For $Ax = b, A \succ 0$, $x_0 = 0, \eta = \frac{1}{\|A\|_2}$, and $x_{k+1} = x_k - \eta(Ax_k - b)$.
Then Theorem:

$$\mathsf{berr}_{A,b}(x_k) \leq \tfrac{1}{k}$$

implying complexity $O(n^2/\epsilon)$ for ϵ backward error.

Richardson iteration complexity is $O(n^2/\epsilon)$ (V surprising to me!)

For $Ax = b, A \succ 0$, $x_0 = 0, \eta = \frac{1}{\|A\|_2}$, and $x_{k+1} = x_k - \eta(Ax_k - b)$.
Then Theorem:

$$\mathsf{berr}_{A,b}(x_k) \le \tfrac{1}{k}$$

implying complexity $O(n^2/\epsilon)$ for ϵ backward error.

Proof: nontrivial but elementary.

$$\begin{aligned} Ax_k - b &= A(x_{k-1} - \eta(Ax_{k-1} - b)) - b \\ &= (I - \eta A)(Ax_{k-1} - b) = -(I - \eta A)^k b. \end{aligned}$$

Also $x_k = (I - \eta A)x_{k-1} + \eta b = \eta \sum_{i=0}^{k-1} (I - \eta A)^i b$. So

$$\mathsf{berr}_{A,b}(x_k) = \frac{\|Ax_k - b\|_2}{\|A\|_2 \|x_k\|_2} = \frac{\|(I - \eta A)^k b\|_2}{\|\sum_{i=0}^{k-1} (I - \eta A)^i b\|_2} \le \left\| (I - \eta A)^k \Big(\sum_{i=0}^{k-1} (I - \eta A)^i \Big)^{-1} \right\|_2$$

$$\le \max_{1 \le j \le n} \frac{(1 - \eta\lambda_j)^k}{\sum_{i=0}^{k-1} (1 - \eta\lambda_j)^i} \le \frac{1}{k}.$$

MINBERR: berr-minimizing Krylov method

Recap: Richardson gives $O(1/k)$ berr. What about Krylov methods?

MINBERR: berr-minimizing Krylov method

Recap: Richardson gives $O(1/k)$ berr. What about Krylov methods?

Can show: regularized CG, MINRES achieve $O((\log k/k)^2)$ berr.

MINBERR: berr-minimizing Krylov method

Recap: Richardson gives $O(1/k)$ berr. What about Krylov methods?

Can show: regularized CG, MINRES achieve $O((\log k/k)^2)$ berr.

Better yet: minimize berr in Krylov subspace!

Call **MINBERR** (precursor in Kasenally 95, Kasenally-Simoncini 97)

$$x_k := \mathsf{argmin}_{x \in \mathsf{Span}(b, Ab, \ldots, A^{k-1}b)} \mathsf{berr}_{A,b}(x)$$

MINBERR: berr-minimizing Krylov method

Recap: Richardson gives $O(1/k)$ berr. What about Krylov methods?
Can show: regularized CG, MINRES achieve $O((\log k/k)^2)$ berr.

Better yet: minimize berr in Krylov subspace!

Call **MINBERR** (precursor in Kasenally 95, Kasenally-Simoncini 97)

$$x_k := \mathsf{argmin}_{x \in \mathsf{Span}(b, Ab, \ldots, A^{k-1}b)} \mathsf{berr}_{A,b}(x)$$

Convergence: $\mathsf{berr}_{A,b}(x_k) \leq \dfrac{3}{k^2-1}$

Proof sketch:

$$\min_{x \in \mathcal{K}_k(A,b)} \mathsf{berr}_{A,b}(x) = \min_{p \in \mathcal{P}_{k-1}} \frac{\|Ap(A)b - b\|_2}{\|p(A)b\|_2} \leq \min_{p \in \mathcal{P}_{k-1}} \|(Ap(A) - I)p(A)^{-1}\|_2$$
$$= \max_i \left| \frac{\lambda_i p(\lambda_i) - 1}{p(\lambda_i)} \right| \leq \max_{x \in [0, \|A\|_2]} \left| x - \frac{1}{p(x)} \right|.$$

Then show RHS$\leq 3/(k^2-1)$ via carefully modified Chebyshev polynomials

MINBERR implementation

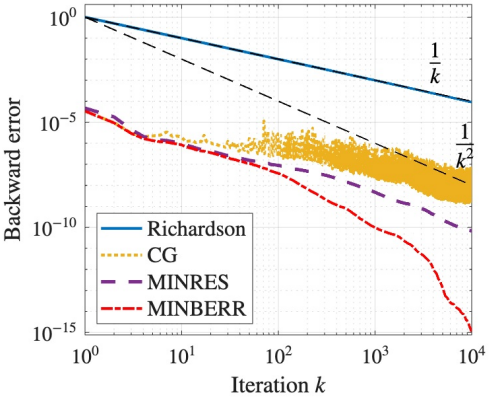
Let $AQ_k = Q_{k+1}T_k$ be Lanczos decomposition.

$$\operatorname{argmin}_{y\in\mathbb{R}^k} \frac{\|AQ_k y - b\|_2}{\|Q_k y\|_2} = \operatorname{argmin}_{y\in\mathbb{R}^k} \frac{\begin{bmatrix}-1 & y^\top\end{bmatrix}\left(\begin{bmatrix}b & AQ_k\end{bmatrix}^\top \begin{bmatrix}b & AQ_k\end{bmatrix}\right)\begin{bmatrix}-1\\ y\end{bmatrix}}{\begin{bmatrix}-1 & y^\top\end{bmatrix}\left(\begin{bmatrix}0 & \\ & I_k\end{bmatrix}^\top \begin{bmatrix}0 & \\ & I_k\end{bmatrix}\right)\begin{bmatrix}-1\\ y\end{bmatrix}}.$$

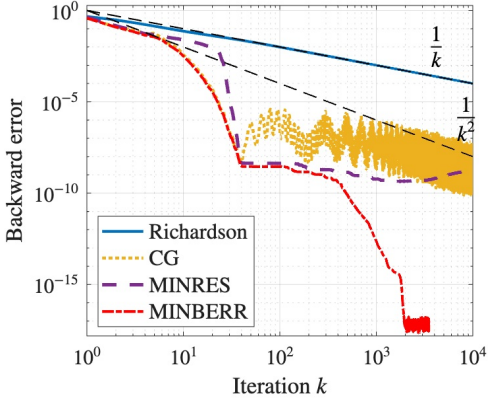
MINBERR solution: smallest eigenpair of generalized eigenproblem.

- Deflate eigenvalue $\lambda = \infty$ and simplifying results in tridiagonal singular value problem $\tilde{T}$.
- Inverse iteration to get smallest singular triplet of $\tilde{T}$; extend analysis in [Kuczyński and Woźniakowski 92]
- In addition to Lanczos, negligible $O(k \log k)$ additional cost needed
- Reorthogonalization makes difference, adding $O(nk^2)$
- Code available on Github

MINBERR Experiments, matrices from SuiteSparse



$n = 16146$ Aerospace model



$n = 19779$ stability analysis

Extension of universal convergence to nonsymmetric $Ax = b$?

Possible approaches:

1. A direct GMRES-style extension of MINBERR (i.e., GMBACK)

2. Richardson on normal equation

3. LSQR-style extension (Krylov for the normal equations)

Extension of universal convergence to nonsymmetric $Ax = b$?

Possible approaches:

1. ~~A direct GMRES-style extension of MINBERR (i.e., GMBACK)~~

   Doesn't work because of the standard counter-example (A orthogonal) where the error stagnates until $k = n$.

2. Richardson on normal equation

3. LSQR-style extension (Krylov for the normal equations)

Extension of universal convergence to nonsymmetric $Ax = b$?

Possible approaches:

1. ~~A direct GMRES-style extension of MINBERR (i.e., GMBACK)~~

   Doesn't work because of the standard counter-example (A orthogonal) where the error stagnates until $k = n$.

2. ~~Richardson on normal equation~~

   Richardson on $A^T Ax = A^T b$ has examples s.t. $berr_{A,\beta}(x_k) \geq \frac{\kappa(A)}{ek}$.

3. LSQR-style extension (Krylov for the normal equations)

MINBERR-NE (MINBERR on normal equations)

MINBERR-NE

Given an $n \times n$ matrix A, vector $\beta \in \mathbb{R}^n$, and $k \geq 1$, define

$$x_k := \mathsf{argmin}_{x \in \mathcal{K}_k(A^T A, A^T \beta)} \mathsf{berr}_{A,\beta}(x).$$

- Still minimize berr, different subspace
- Bidiagonalization-based decomposition, takes $2k \cdot \mathcal{T}_{\mathsf{MatVec}} + O(nk)$

MINBERR-NE (MINBERR on normal equations)

MINBERR-NE

Given an $n \times n$ matrix A, vector $\beta \in \mathbb{R}^n$, and $k \geq 1$, define

$$x_k := \text{argmin}_{x \in \mathcal{K}_k(A^TA, A^T\beta)} \text{berr}_{A,\beta}(x).$$

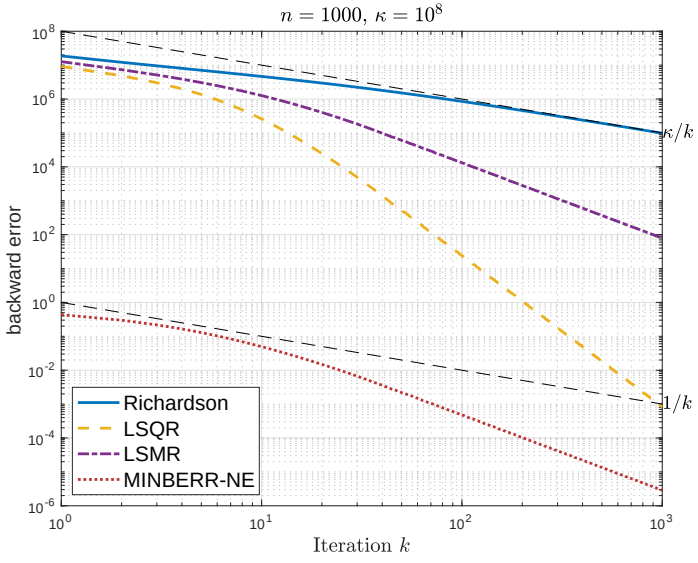
- Still minimize berr, different subspace
- Bidiagonalization-based decomposition, takes $2k \cdot \mathcal{T}_{\text{MatVec}} + O(nk)$

Theorem (Nearly-universal convergence of MINBERR-NE)

For all $k \geq 2$, the MINBERR-NE iterate sequence satisfies:

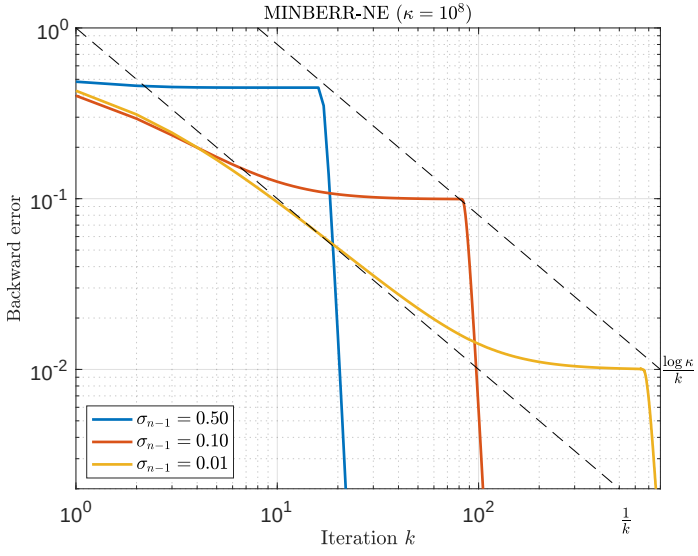
$$\textit{berr}_{A,\beta}(x_k) \leq \frac{3 \log \kappa(A)}{k}.$$

Toy experiment



$\kappa(A) = 10^8$, geometric $\sigma_i(A)$, b: nearly aligned with smallest singvec

Hard case for MINBERR-NE



$A = U\Sigma V^\top$, $\sigma_i(A)$ geom decay, except $\sigma_n \ll \sigma_{n-1}$. $\beta = U \cdot [1, \ldots, 1, \sqrt{n}]^\top$

Random perturbation to the rescue

Perturbed MINBERR-NE

Given $A \in \mathbb{R}^{n\times n}$, $\beta \in \mathbb{R}^n$, $k \geq 1$, $\epsilon \geq 0$, and a Gaussian $G \in \mathbb{R}^{n\times n}$,

$$\tilde{x}_k \ := \text{argmin}_{x\in\mathcal{K}_k(\tilde{A}^\top\tilde{A},\tilde{A}^\top\beta)} \text{berr}_{\tilde{A},\beta}(x), \quad \text{where} \quad \tilde{A} = A + \frac{\epsilon\|A\|_2}{3\sqrt{n}}G.$$

- G perturbation only adds $\approx \epsilon$ backward error.
- $\kappa_2(\tilde{A}) = O(n/\epsilon)$ with high probability. [Sankar-Spielman-Teng 06]

Random perturbation to the rescue

Perturbed MINBERR-NE

Given $A \in \mathbb{R}^{n\times n}$, $\beta \in \mathbb{R}^n$, $k \geq 1$, $\epsilon \geq 0$, and a Gaussian $G \in \mathbb{R}^{n\times n}$,

$$\tilde{x}_k \ := \text{argmin}_{x\in\mathcal{K}_k(\tilde{A}^\top\tilde{A},\tilde{A}^\top\beta)} \text{berr}_{\tilde{A},\beta}(x), \quad \text{where} \quad \tilde{A} = A + \tfrac{\epsilon\|A\|_2}{3\sqrt{n}}G.$$

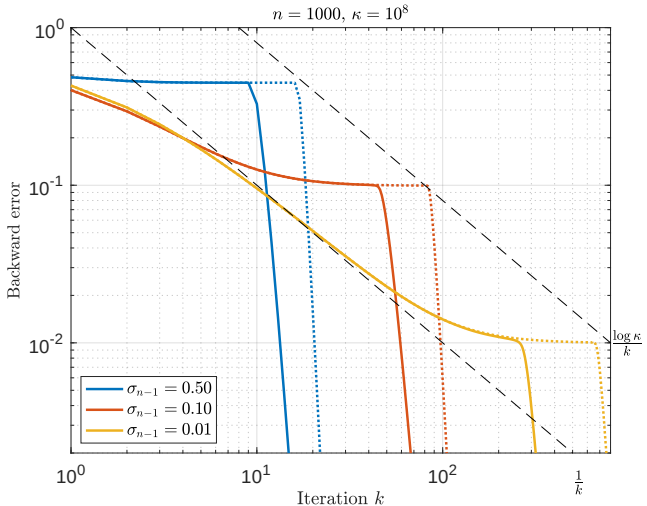
- G perturbation only adds $\approx \epsilon$ backward error.
- $\kappa_2(\tilde{A}) = O(n/\epsilon)$ with high probability. [Sankar-Spielman-Teng 06]

Theorem (Quasi-Universal convergence of Perturbed MINBERR-NE)

With probability $1 - \delta - e^{-n/2}$, *Perturbed MINBERR-NE satisfies:*

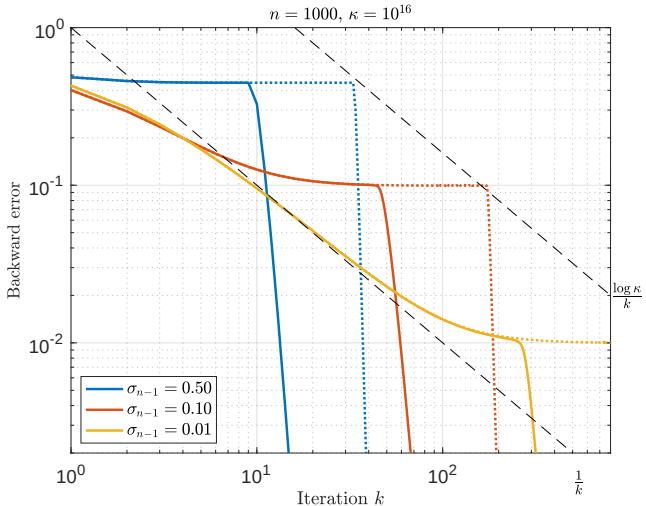
$$\textit{berr}_{A,\beta}(\tilde{x}_k) \ \leq \frac{4\log(10n/\epsilon\delta)}{k} + \epsilon.$$

MINBERR-NE hard case again



$A = U\Sigma V^\top$, $\sigma_i(A)$ geom decay, except $\sigma_n \ll \sigma_{n-1}$. $\beta = U \cdot [1, ..., 1, \sqrt{n}]^\top$

MINBERR-NE hard case again



$A = U\Sigma V^\top$, $\sigma_i(A)$ geom decay, except $\sigma_n \ll \sigma_{n-1}$. $\beta = U \cdot [1, ..., 1, \sqrt{n}]^\top$

Complexity of MINBERR-NE

Theorem

With probability $\geq 1-\delta$*, perturbed MINBERR-NE gets berr*$\leq \epsilon$ *in:*

$$O(n^2 \log(n/\delta)/\epsilon) \text{ time} \quad \text{and} \quad O(\log(n)/\epsilon) \text{ matrix-vector products.}$$

Complexity of MINBERR-NE

Theorem

With probability $\geq 1-\delta$*, perturbed MINBERR-NE gets berr*$\leq \epsilon$ *in:*

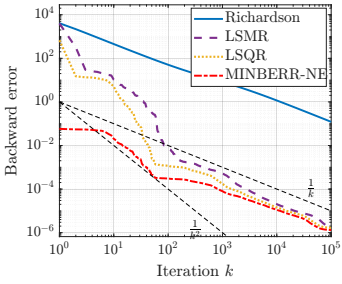
$$O(n^2 \log(n/\delta)/\epsilon) \text{ time} \quad \text{and} \quad O(\log(n)/\epsilon) \text{ matrix-vector products.}$$

Weaker result than PSD case:

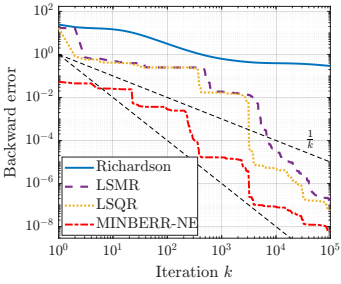
- $k = O(\log(n/\delta)/\epsilon)$ rather than $O(1/\sqrt{\epsilon})$
- Not truly universal convergence, because of $\log n$ factor.
- Gaussian perturbation requires dense matrix-vector products.

But comes pretty close!

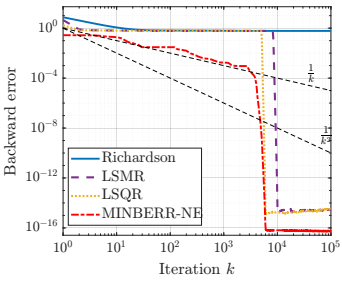
Some experiments on real data



`sherman3` data:
non-symmetric, $n = 5005$
(Fluid Dynamics)



`bayer03` data:
non-symmetric, $n = 6747$
(Chemical Simulation)



`cyl6` data:
non-symmetric, $n = 13681$
(Shell Mechanics)

Summary

We can solve any $n \times n$ PSD linear system in $O(n^2/\epsilon)$ time.

We can solve any $n \times n$ linear system in $O(n^2 \log(n)/\epsilon)$ time.

- Backward error converges **universally** for $Ax = b$, $A \succeq 0$.
  - k steps of Richardson iteration gives berr $\leq \frac{1}{k}$.
  - k steps of MINBERR gives berr $\leq \frac{1}{k^2}$.
- For ϵ berr, MINBERR needs $k \lesssim \frac{1}{\sqrt{\epsilon}}$ iterations.
  - In single precision $\epsilon \approx 2 \times 10^{-8}$, this is $k \approx 5000$.
- MINBERR-NE: Extension to general $n \times n$ linear systems

Summary

We can solve any $n \times n$ PSD linear system in $O(n^2/\epsilon)$ time.

We can solve any $n \times n$ linear system in $O(n^2 \log(n)/\epsilon)$ time.

- Backward error converges **universally** for $Ax = b$, $A \succeq 0$.
  - k steps of Richardson iteration gives berr $\leq \frac{1}{k}$.
  - k steps of MINBERR gives berr $\leq \frac{1}{k^2}$.
- For ϵ berr, MINBERR needs $k \lesssim \frac{1}{\sqrt{\epsilon}}$ iterations.
  - In single precision $\epsilon \approx 2 \times 10^{-8}$, this is $k \approx 5000$.
- MINBERR-NE: Extension to general $n \times n$ linear systems

Open problems:

- Is universal convergence possible for $A \neq A^T$?
- MINBERR behavior in finite precision
- **Preconditioned** MINBERR: implementation, convergence theory?
- Least squares problems? $f(A)b$?

References I

N. Amsel, T. Chen, F. D. Keles, D. Halikias, C. Musco, and C. Musco.
Fixed-sparsity matrix approximation from matrix-vector products.
SIAM J. Matrix Anal. Appl., 47(2):483–511, 2026.

O. Balabanov and L. Grigori.
Randomized gram–schmidt process with application to GMRES.
SIAM J. Sci. Comput., 44(3):A1450–A1474, 2022.

N. Boullé, D. Halikias, S. E. Otto, and A. Townsend.
Operator learning without the adjoint.
Journal of Machine Learning Research, 25(364):1–54, 2024.

A. Bucci, Y. Nakatsukasa, and T. Park.
Numerical stability of the nyström method.
arXiv preprint arXiv:2511.15583, 2025.

E. Carson and I. Daužickaitė.
Single-pass nyström approximation in mixed precision.
SIAM J. Matrix Anal. Appl., 45(3):1361–1391, 2024.

References II

M. P. Connolly, N. J. Higham, and S. Pranesh.
Randomized low rank matrix approximation: Rounding error analysis and a mixed precision algorithm.
(2022.10), 2022.

M. Dereziński, E. N. Epperly, D. Needell, and A. Xue.
Numerical instabilities in the kaczmarz method and stabilization by iterative refinement.
arXiv preprint arXiv:2605.17185, 2026.

M. Dereziński, Y. Nakatsukasa, and E. Rebrova.
Towards universal convergence of backward error in linear system solvers.
arXiv preprint arXiv:2604.16075, 2026.

E. N. Epperly, A. Greenbaum, and Y. Nakatsukasa.
Stable algorithms for general linear systems by preconditioning the normal equations.
Numerische Mathematik, pages 1–28, 2026.

N. J. Higham.
Accuracy and Stability of Numerical Algorithms.
SIAM, Philadelphia, PA, USA, second edition, 2002.

References III

J. Levitt and P.-G. Martinsson.
Linear-complexity black-box randomized compression of rank-structured matrices.
SIAM J. Sci. Comput., 46(3):A1747–A1763, 2024.

M. Meier, Y. Nakatsukasa, A. Townsend, and M. Webb.
Are sketch-and-precondition least squares solvers numerically stable?
SIAM J. Matrix Anal. Appl., 45(2):905–929, 2024.

Y. Nakatsukasa.
Fast and stable randomized low-rank matrix approximation.
arXiv preprint arXiv:2009.11392, 2020.

Y. Nakatsukasa and J. A. Tropp.
Fast and accurate randomized algorithms for linear systems and eigenvalue problems.
SIAM J. Matrix Anal. Appl., 45(2):1183–1214, 2024.

T. Park and Y. Nakatsukasa.
Accuracy and stability of cur decompositions with oversampling.
SIAM J. Matrix Anal. Appl., 46(1):780–810, 2025.

References IV

V. Rokhlin and M. Tygert.
A fast randomized algorithm for overdetermined linear least-squares regression.
Proc. Natl. Acad. Sci., 105(36):13212–13217, 2008.

Y. Saad and M. H. Schultz.
GMRES - A generalized minimal residual algorithm for solving nonsymmetric linear systems.
SIAM J. Sci. Stat. Comp., 7(3):856–869, 1986.

T. Strohmer and R. Vershynin.
A randomized Kaczmarz algorithm with exponential convergence.
J. Fourier Anal. Appl., 15(2):262, 2009.

L. N. Trefethen and D. Bau.
Numerical Linear Algebra.
SIAM, Philadelphia, 1997.

J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher.
Practical sketching algorithms for low-rank matrix approximation.
SIAM J. Matrix Anal. Appl., 38(4):1454–1485, 2017.

References V

M. Udell and A. Townsend.
Why are big data matrices approximately low rank?
SIAM Journal on Mathematics of Data Science, 1(1):144–160, 2019.